به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین چهارم

نام و نام خانوادگی	محمد مهدی کعبی – محمد امانلو
شماره دانشجویی	۸۱۰۱۰۲۵۶۱ - ۸۱۰۱۰۰۰۸۴
تاریخ ارسال گزارش	۱۴۰۳,۱۰.۰۱

فهرست

پرسش 1.تشخیص هرزنامه ۱

۱-۱. مجموعه داده ۱

۱-۲ : پیش پردازش داده ها ۲

۱-۳ نمایش ویژگی ها ۳

4-1 ساخت مدل ۶

۱-۵ ارزیابی ۲۱

پاسخ ۲ – پیش‌بینی قیمت نفت خام ۲۴

۲-۱. مقدمه ۲۴

۲-۲. مجموعه دادگان و آماده‌سازی ۲۴

۲-۳. پیاده‌سازی مدل‌ها ۲۹

۲-۴. ARIMA ۳۹

شکل‌ها

شکل ۱توزیع کلاس ها در ستون لیبل ۱
شکل ۲ تغییرات Loss و دقت مدل ها در ایپاک های مختلف ۹
شکل ۳ نمودار ROC بدست آمده از مدل ۱۱
شکل ۴ Confusion matrix بدست آمده از مدل ۱۲
شکل ۵ نمودار دقت و خطا در داده های آموزشی و ارزیابی ۱۳
شکل ۶ نمودار ROC| برای مدل CNN ۱۵
شکل ۷ ماتریس سردرگمی برای مدل CNN ۱۶
شکل ۸ نمودار دقت و خطا در داده های آموزشی و ارزیابی برای مدل LSTM ۱۷
شکل ۹ نمودار ROC برای مدل LSTM ۱۸
شکل ۱۰ ماتریس سردرگمی برای مدل LSTM ۱۹
شکل ۱۱ نحوه پر کردن داده های از دست رفته ۲۵
شکل ۱۲هیستوگرام توزیع قیمت ۲۸
شکل ۱۳ مقایسه پیش‌بینی های GRU با مقادیر واقعی ۳۲
شکل ۱۴مقایسه پیش‌بینی‌های Bi-LSTM با مقادیر واقعی ۳۲
شکل ۱۵ مقایسه پیش‌بینی‌های LSTM با مقادیر واقعی ۳۳
شکل ۱۶ مقایسه پیش‌بینی‌های GRU با مقادیر اسکیل شده ۳۳
شکل ۱۷ مقایسه پیش‌بینی‌های Bi-LSTM با مقادیر اسکیل شده ۳۳
شکل ۱۸مقایسه مقادیر پیش‌بینی شده LSTM با مقادیر واقعی اسکیل شده ۳۴
شکل ۱۹ فرمول محاسبه R2Score ۳۶
شکل ۲۰ فرمول محاسبه RMSE ۳۶
شکل ۲۱ فرمول محاسبه MAE ۳۷
شکل ۲۲فرمول محاسبه MAPE ۳۷
شکل ۲۳ دقت مدل های یادگیری عمیق در حالات مختلف ۳۷
شکل ۲۴انجام تست ADF ۴۶
شکل ۲۵پارامترهای انتخاب شده برای ARIMA ۴۷
شکل ۲۶ دقت مدل ARIMA ۴۹
شکل ۲۷ مقایسه خروجی های مدل ARIMA با مقادیر واقعی ۴۹

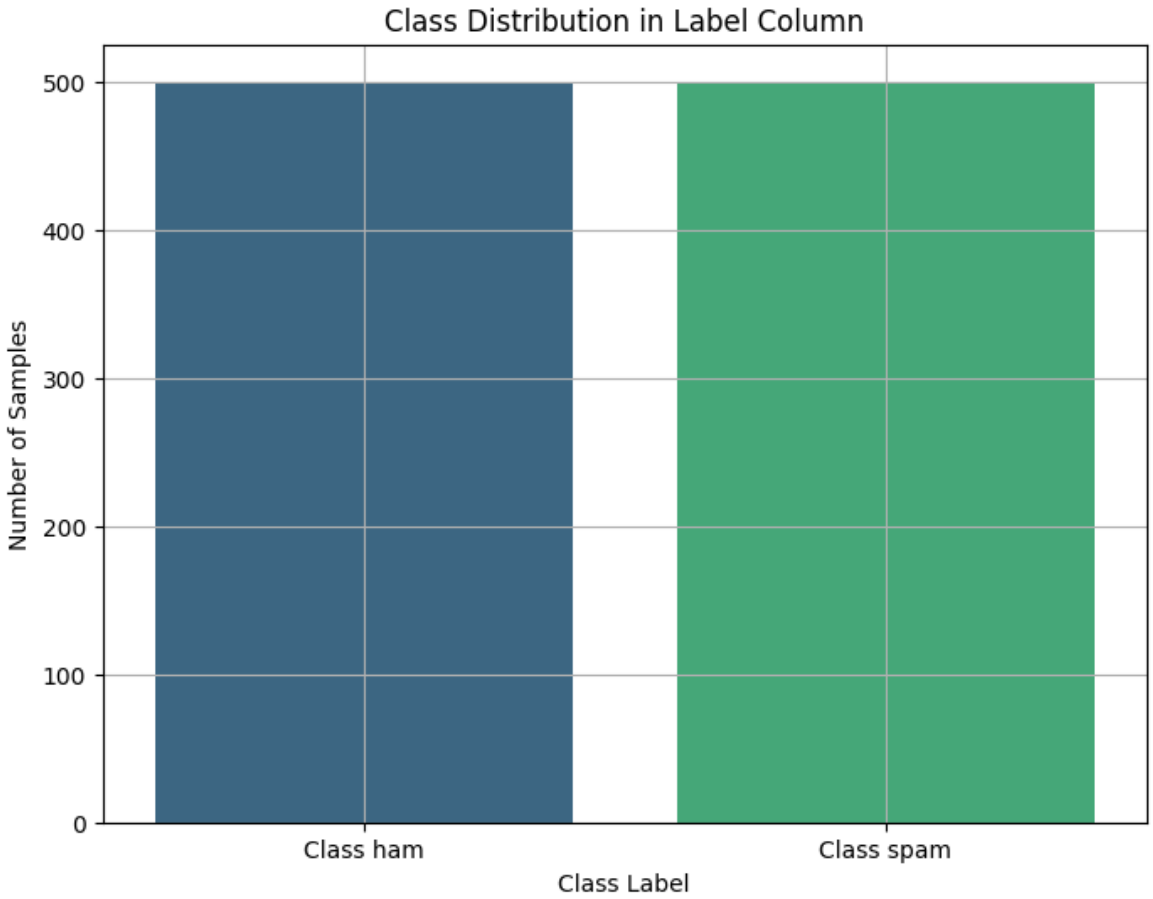
جدول‌ها

جدول1 جدول معماری مدل های LSTM-CNN ۷
جدول2 مقادیر هایپرپارامترهای تنظیم شده ۸
جدول3 معماری مدل CNN ۱۳
جدول4 جدول معماری مدل LSTM ۱۷
جدول5 مقایسه نتیجه همه مدل‌ها ۲۱
جدول6 مقایسه نتایج بدست آمده از مدل های مختلف ۵۲

پرسش 1.تشخیص هرزنامه

۱-۱. مجموعه داده

ابتدا کدی برای بررسی توزیع کلاس‌ها در مجموعه داده ایمیل‌ها نوشتیم. این کد با استفاده از کتابخانه‌های Pandas، Matplotlib و Seaborn داده‌ها را بارگذاری کرده و تعداد نمونه‌های هر کلاس را در ستون label شمارش کرد. سپس یک نمودار میله‌ای برای نمایش تصویری توزیع این کلاس‌ها رسم شد.



شکل ۱توزیع کلاس ها در ستون لیبل

نمودار میله‌ای فوق، تعداد نمونه‌های هر کلاس را در مجموعه داده نشان می‌دهد.

کلاس "ham" شامل ۵۰۰ نمونه است که بیانگر ایمیل‌های غیر اسپم (عادی) می‌باشد.

کلاس "spam" نیز شامل ۵۰۰ نمونه است که نشان‌دهنده ایمیل‌های اسپم (ناخواسته) است.

بررسی این نمودار نشان می‌دهد که مجموعه داده متوازن است؛ به این معنا که تعداد نمونه‌های دو کلاس **ham** و **spam** برابر هستند. این تعادل می‌تواند تأثیر مثبتی روی عملکرد مدل یادگیری ماشین داشته باشد زیرا مدل نیازی به جبران عدم توازن داده‌ها ندارد.

۱-۲ : پیش پردازش داده ها

در این بخش، فرآیند پیش‌پردازش داده‌ها برای آماده‌سازی مجموعه ایمیل‌ها جهت مدل‌سازی ارائه شده است. این فرآیند شامل پاک‌سازی، نرمال‌سازی و ساده‌سازی متن‌ها با هدف بهبود عملکرد مدل‌های یادگیری ماشین است. در ادامه، مراحل انجام‌شده به تفصیل توضیح داده می‌شود

بارگذاری داده‌ها:

داده‌ها از مسیر content/emails.csv/ خوانده شده‌اند.

بررسی شده که ستون‌های text و label در مجموعه داده موجود باشند.

این ستون‌ها به ترتیب شامل محتوای ایمیل‌ها و برچسب کلاس (spam یا ham) هستند.

نرمال‌سازی متن:

برای یکدست‌سازی زبان فارسی، متن‌ها با استفاده از کتابخانه Hazm نرمال‌سازی شدند.

این نرمال‌سازی شامل اصلاح نیم‌فاصله‌ها، تبدیل اعداد فارسی به انگلیسی و استانداردسازی حروف فارسی است.

حذف لینک‌ها (URLs):

تمام لینک‌های موجود در متن که ممکن است اطلاعات غیرمرتبط ارائه دهند، حذف شدند.

حذف آدرس‌های ایمیل:

آدرس‌های ایمیل که می‌توانند منحصربه‌فرد و بی‌ارتباط به محتوای ایمیل باشند، حذف شدند.

حذف شماره‌های تلفن:

الگوهای مربوط به شماره‌های تلفن، شامل اعداد با فاصله یا بدون فاصله، حذف شدند.

کاهش تکرار حروف:

حروف تکراری که ممکن است در زبان محاوره یا تأکیدهای احساسی مانند "عاااالی" استفاده شوند، به یک حرف تبدیل شدند (مانند "عالی").

حذف کلمات توقف (Stopwords):

کلمات پرکاربرد اما کم‌ارزش از نظر معنایی مانند "و"، "که"، "به"، "از" حذف شدند.

این کار باعث کاهش نویز و تمرکز بر کلمات کلیدی می‌شود.

توکن‌سازی (Tokenization):

متن‌ها به کلمات جداگانه شکسته شدند تا پردازش‌های بعدی مانند تعبیه کلمات (Word Embedding) آسان‌تر انجام شوند.

۳-۱ نمایش ویژگی ها

در این بخش، فرآیند استخراج ویژگی‌ها از داده‌های پردازش‌شده با استفاده از مدل قدرتمند ParsBERT انجام شد. هدف از این مرحله، تبدیل متن‌های پردازش‌شده به بردارهای عددی قابل استفاده برای مدل‌های یادگیری ماشین بود. ابتدا داده‌ها بارگذاری شدند و برای اطمینان از کیفیت آن‌ها، مقادیر گم‌شده از مجموعه حذف گردیدند. سپس با بهره‌گیری از مدل از پیش آموزش‌دیده ParsBERT، ویژگی‌های معنایی جملات استخراج شد.

پس از استخراج ویژگی‌ها، با توجه به اینکه بردارهای تولیدشده ابعاد بالایی داشتند، برای جلوگیری از پیچیدگی محاسبات و خطر Overfitting، از روش Truncated SVD برای کاهش ابعاد به ۱۲۰ مؤلفه اصلی استفاده شد. این مرحله کمک کرد تا اطلاعات کلیدی حفظ شوند و نویزهای احتمالی حذف گردند.

سوال

ابعاد پیش‌فرض بردار تعبیه در ParsBERT چقدر است؟

ابعاد پیش‌فرض بردار تعبیه (**Embedding Dimension**) در مدل **ParsBERT** برابر با **768**است.

این مقدار مطابق با معماری **BERT-base** است که دارای:

- **12 لایه(Transformer Layers)**
- **12 هد توجه(Attention Heads)**
- **768 بعد برای بردارهای تعبیه شده**

هر توکن ورودی در مدل ParsBERT به یک بردار ۷۶۸ بُعدی تبدیل می‌شود که شامل اطلاعات معنایی و موقعیتی آن توکن در متن است. این بردارها سپس می‌توانند برای انجام وظایفی مانند **طبقه‌بندی متن، تحلیل احساسات** یا **تشخیص نهادهای نامدار (NER)** استفاده شوند.

تعداد ابعاد این بردار بیانگر چیست؟

تعداد ابعاد بردار تعبیه **768) بعد** (در مدل **ParsBERT**بیانگر مقدار اطلاعات معنایی و نحوی است که هر توکن (کلمه یا زیرکلمه) در متن ورودی نمایش می‌دهد. این ابعاد به‌طور خلاصه، فضای برداری چندبعدی را ایجاد می‌کنند که در آن ویژگی‌های زبانی مختلف توکن‌ها کدگذاری می‌شوند.

تحلیل مفهوم ابعاد بردار تعبیه:

۱. **نمایش معنایی:(Semantic Representation)**
   - هر بعد از بردار ممکن است بخشی از معنای کلمه را نشان دهد. به‌عنوان مثال، برخی ابعاد می‌توانند ارتباط کلمه با احساسات (مثبت یا منفی) را نشان دهند و برخی دیگر موضوع یا زمینه استفاده کلمه را منعکس کنند.

۲. **ارتباط نحوی:(Syntactic Representation)**
   - برخی ابعاد اطلاعات نحوی مانند نقش گرامری (فاعل، مفعول، فعل) یا ساختار جمله را ذخیره می‌کنند.

۳. **وابستگی‌های متنی:(Contextual Representation)**
   - مدلParsBERT ، مانندBERT ، **زمینه‌محور (Contextual)** است؛ به این معنا که بردار هر کلمه به کلمات اطراف آن بستگی دارد.

۴. **نمایش چندمنظوره:(Multi-faceted Representation)**

- ابعاد مختلف بردار به مدل اجازه می‌دهند هم‌زمان چندین ویژگی را ذخیره کند؛ از جمله جنسیت، زمان، حالت مجهول یا معلوم، و حتی روابط مفهومی مانند مترادف‌ها و متضادها.

چرا تعداد ابعاد برابر ۷۶۸ انتخاب شده است؟

- این تعداد ابعاد از معماری اصلی **BERT-base** گرفته شده است که تعادلی بین **کارایی** و **دقت** برقرار می‌کند.
- تعداد بیشتر ابعاد (مثلاً ۱۰۲۴ در **BERT-large**) اطلاعات بیشتری ذخیره می‌کند اما به منابع محاسباتی بیشتری نیاز دارد.
- ابعاد کمتر ممکن است باعث از دست رفتن اطلاعات کلیدی و کاهش دقت مدل شود.

مفهوم بردار تعبیه را توضیح دهید و بیان کنید کدام کلمات موجود در مجموعه داده ممکن است تعبیه‌ای نزدیک به هم داشته باشند؟

بردار تعبیه یا Embedding، یک نمایش عددی از کلمات در قالب بردارهایی در فضای چندبعدی است که روابط معنایی و نحوی بین کلمات را حفظ می‌کند. این تکنیک به‌عنوان ابزاری قدرتمند در پردازش زبان طبیعی (NLP) استفاده می‌شود و هدف آن این است که کلمات با معانی مشابه، در فضای برداری به یکدیگر نزدیک‌تر باشند.

در مجموعه داده ایمیل‌های ما، کلماتی که ممکن است بردار تعبیه مشابهی داشته باشند عبارت‌اند از:

کلمات هم‌معنی یا مترادف:

جایزه و هدیه (هر دو مرتبط با پاداش)

فروش و تخفیف (مرتبط با تبلیغات تجاری)

کلمات دارای زمینه معنایی مشابه:

بانک و وام (هر دو مرتبط با امور مالی)

ثبت‌نام و عضویت (مرتبط با فرآیند ثبت اطلاعات)

کلمات مرتبط با تبلیغات یا پیشنهادات خاص:

رایگان و تخفیف

جایزه و قرعه‌کشی

کلمات مرتبط با ارتباطات شخصی:

سلام و دوست

محترم و گرامی

4-1 ساخت مدل

در ابتدا داده ها را به صورت زیر تقسیم کرده ایم

٪۷۰ برای آموزش (Training):

شامل ۵۶۰ نمونه.

٪۲۰ از آموزش برای اعتبارسنجی (Validation):

شامل ۱۴۰ نمونه.

٪۳۰ برای تست (Testing):

شامل ۳۰۰ نمونه.

مدل CNN-LSTM

در این قسمت یک مدل ترکیبی CNN-LSTM برای طبقه‌بندی ایمیل‌ها به دو کلاس spamو ham طراحی و آموزش داده شده است. این مدل با ترکیب ویژگی‌های مکانی از طریق شبکه عصبی پیچشی (CNN) و یادگیری وابستگی‌های ترتیبی با شبکه حافظه کوتاه‌مدت طولانی (LSTM) بهینه شده است.

جدول معماری مدل CNN-LSTM

جدول**1** جدول معماری مدل های **LSTM-CNN**

Layer (Type)	Output Shape	Param #	Description
Embedding (Embedding)	(None, 150, 64)	384,000	تعبیه کلمات به بردارهای ۶۴ بعدی با اندازه واژگان ۶۰۰۰.
Conv1D (Convolutional)	(None, 148, 64)	12,352	فیلترهای ۱ بعدی با اندازه کرنل ۳ و تعداد ۶۴ فیلتر برای استخراج ویژگی‌ها.
MaxPooling1D (Pooling)	(None, 74, 64)	0	کاهش ابعاد ویژگی‌ها با اندازه پنجره ۲ برای تمرکز روی اطلاعات کلیدی.
Dropout (Regularization)	(None, 74, 64)	0	خاموش کردن ۵۰ درصد نورون‌ها برای جلوگیری از Overfitting.
LSTM (Recurrent)	(None, 64)	33,024	یادگیری وابستگی‌های ترتیبی در داده‌ها با ۶۴ واحد حافظه (Cells).

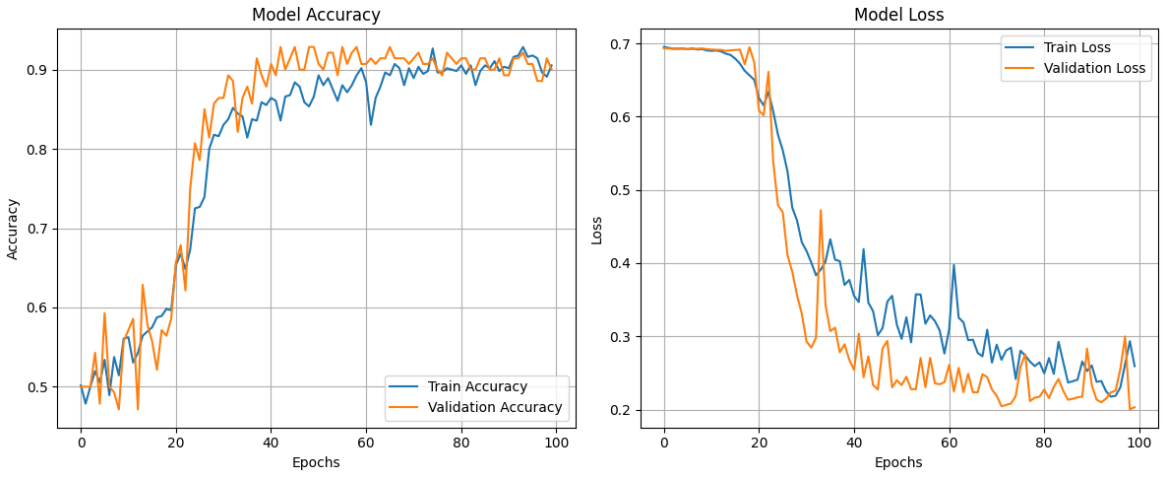
Dense (Output)	(None, 1)	65	خروجی با یک نورون و تابع فعال‌سازی sigmoid برای طبقه‌بندی دودویی.

مقادیر هایپر پارامتر های تنظیم شده

جدول**2** مقادیر هایپرپارامترهای تنظیم شده

هایپرپارامتر	مقدار تنظیم‌شده
Learning Rate	0.001
Batch Size	64
Epochs	100
Optimizer	Adam

در این مرحله، مدل طراحی‌شده CNN-LSTM به مدت ۱۰۰ دوره آموزشی (Epoch) بر روی داده‌های پردازش‌شده ایمیل‌ها آموزش داده شد. نمودارهای پایین نشان‌دهنده روند تغییر دقت (Accuracy) و خطا (Loss) در طول فرآیند آموزش برای مجموعه‌های آموزشی (Train) و اعتبارسنجی (Validation) هستند.



شکل ۲ تغییرات **Loss** و دقت مدل ها در ایپاک های مختلف

در نتیجه مدل توانسته است با افزایش ایپاک‌ها، ویژگی‌های پیچیده‌تری را یاد بگیرد که باعث بهبود دقت شده است.

ارزیابی مدل **CNN-LSTM**

معیار های ارزیابی مدل اینگونه شد

Accuracy: 0.8467

Precision: 0.8250

Recall: 0.8800

F1 Score: 0.8516

Accuracy (دقت کلی):

مقدار ۸۴٫۶۷٪ نشان‌دهنده توانایی کلی مدل در پیش‌بینی صحیح نمونه‌های Spam و Ham است.

این مقدار بیانگر این است که مدل به‌خوبی توانسته است اکثریت نمونه‌ها را به‌درستی طبقه‌بندی کند.

Precision (دقت پیش‌بینی مثبت):

مقدار ۸۲٫۵۰٪ نشان می‌دهد که از میان نمونه‌هایی که مدل به‌عنوان Spam پیش‌بینی کرده، ۸۲٫۵٪ واقعاً Spam بوده‌اند.

این معیار به‌خصوص در سناریوهایی اهمیت دارد که نرخ خطای مثبت کاذب (False Positive) باید پایین باشد، مانند شناسایی اسپم که نباید ایمیل‌های معتبر به‌اشتباه اسپم شناخته شوند.

Recall (بازخوانی):

مقدار ۸۸٫۰۰٪ بیانگر توانایی مدل در شناسایی درست همه نمونه‌های Spam است.

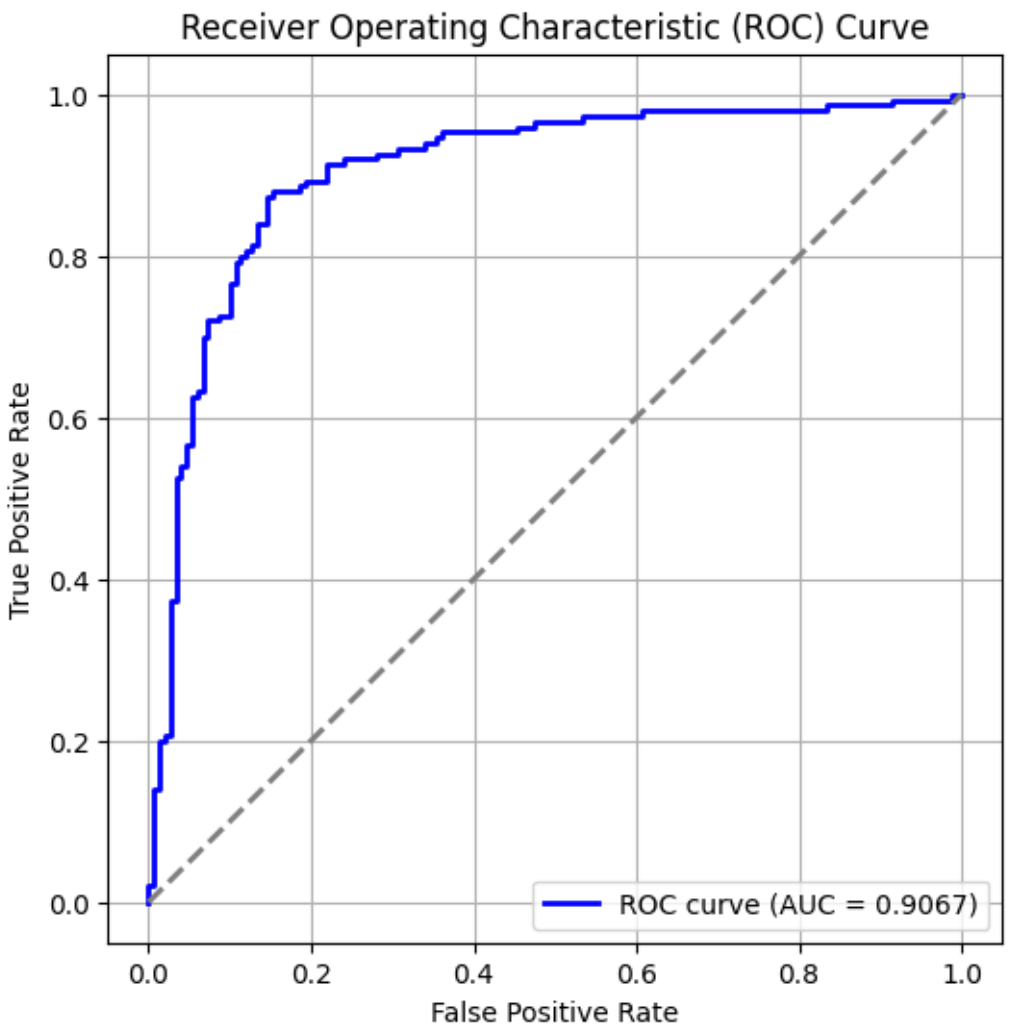
این مقدار بالا نشان می‌دهد که مدل موارد اسپم را به‌خوبی پوشش داده و تعداد کمی از آن‌ها را از دست داده است.

F1-Score (تعادل بین Precision و Recall):

مقدار ۸۵٫۱۶٪ به‌عنوان میانگین هماهنگ بین دقت و بازخوانی، عملکرد متعادلی را نمایش می‌دهد.

این معیار تأیید می‌کند که مدل هم در شناسایی موارد مثبت و هم در جلوگیری از خطای مثبت کاذب عملکرد مناسبی دارد.

نمودار ROC و معیار AUC:



شکل ۳ نمودار **ROC** بدست آمده از مدل

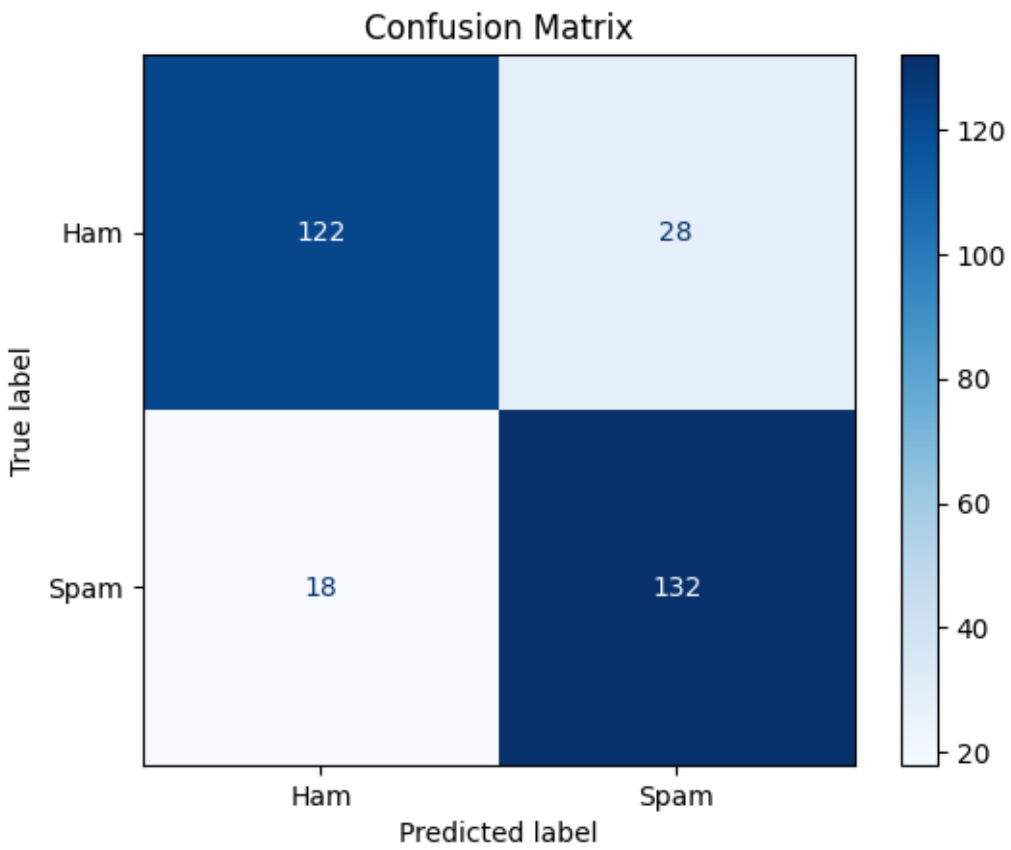
نمودار ROC Curve نشان می‌دهد که مدل در جداسازی کلاس‌ها عملکرد مناسبی دارد. مقدار AUC = 0.9067 تأیید می‌کند که مدل توانایی بالایی در تفکیک کلاس‌های Spam و Ham دارد

تحلیل AUC:

مقدار نزدیک به ۱,۰ ایده‌آل است.

مقدار بالای ۰,۹۰ نشان می‌دهد که مدل در اکثر موارد قادر است پیش‌بینی‌های درستی انجام دهد و نسبت خطاها پایین است.

ماتریس سردرگمی (Confusion Matrix)



شکل ۴ **Confusion matrix** بدست آمده از مدل

تحلیل ماتریس:

True Positives (۱۳۲): تعداد ایمیل‌های Spam که به‌درستی شناسایی شده‌اند.

True Negatives (۱۲۲): تعداد ایمیل‌های Ham که به‌درستی شناسایی شده‌اند.

False Positives (۲۸): ایمیل‌های Ham که به اشتباه به‌عنوان Spam شناسایی شده‌اند.

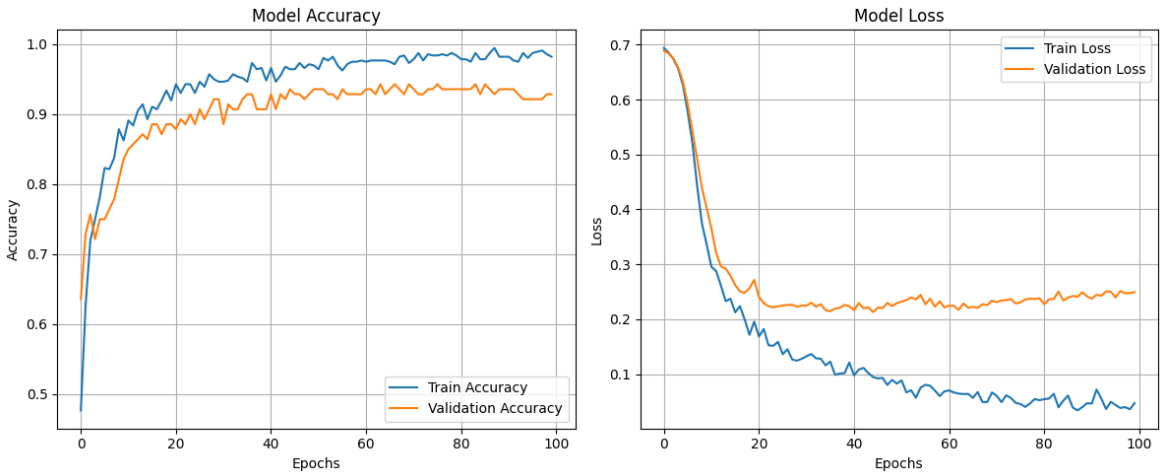
False Negatives (۱۸): ایمیل‌های Spam که به اشتباه به‌عنوان Ham شناسایی شده‌اند.

مدل CNN

جدول3 معماری مدل CNN

Layer (Type)	Output Shape	Param #
Embedding (Embedding)	(None, 150, 64)	384,000
Conv1D (Convolutional)	(None, 148, 64)	12,352
MaxPooling1D (Pooling)	(None, 74, 64)	0
Dropout (Dropout)	(None, 74, 64)	0
Flatten (Flatten)	(None, 4736)	0
Dense (Output)	(None, 1)	4737

نمودار دقت و خطا داده های آموزشی و ارزیابی



شکل ۵ نمودار دقت و خطا در داده های آموزشی و ارزیابی

مدل CNN طراحی‌شده عملکرد قدرتمندی در استخراج ویژگی‌ها و شناسایی الگوهای متنی از خود نشان داده است. با دستیابی به دقت بالای ۹۴٪ در اعتبارسنجی و خطای پایین در آموزش، این مدل پتانسیل بالایی برای تشخیص ایمیل‌های Spam و Ham دارد. هرچند که اندکی Overfitting مشاهده می‌شود، این مشکل با به‌کارگیری روش‌های بهینه‌سازی و تنظیم‌های بیشتر قابل کنترل است.

ارزیابی مدل CNN

معیار های ارزیابی مدل اینگونه شد

Accuracy: 0.8767

Precision: 0.8897

Recall: 0.8600

F1 Score: 0.8746

Accuracy (دقت کلی):

مقدار ۸۷٫۶۷٪ نشان می‌دهد که مدل به‌طور کلی در پیش‌بینی صحیح کلاس‌ها عملکرد مطلوبی داشته است.

Precision (دقت پیش‌بینی مثبت):

مقدار ۸۸٫۹۷٪ بیانگر این است که از میان ایمیل‌هایی که مدل به‌عنوان Spam پیش‌بینی کرده، ۸۸٫۹۷٪ واقعاً اسپم بوده‌اند.

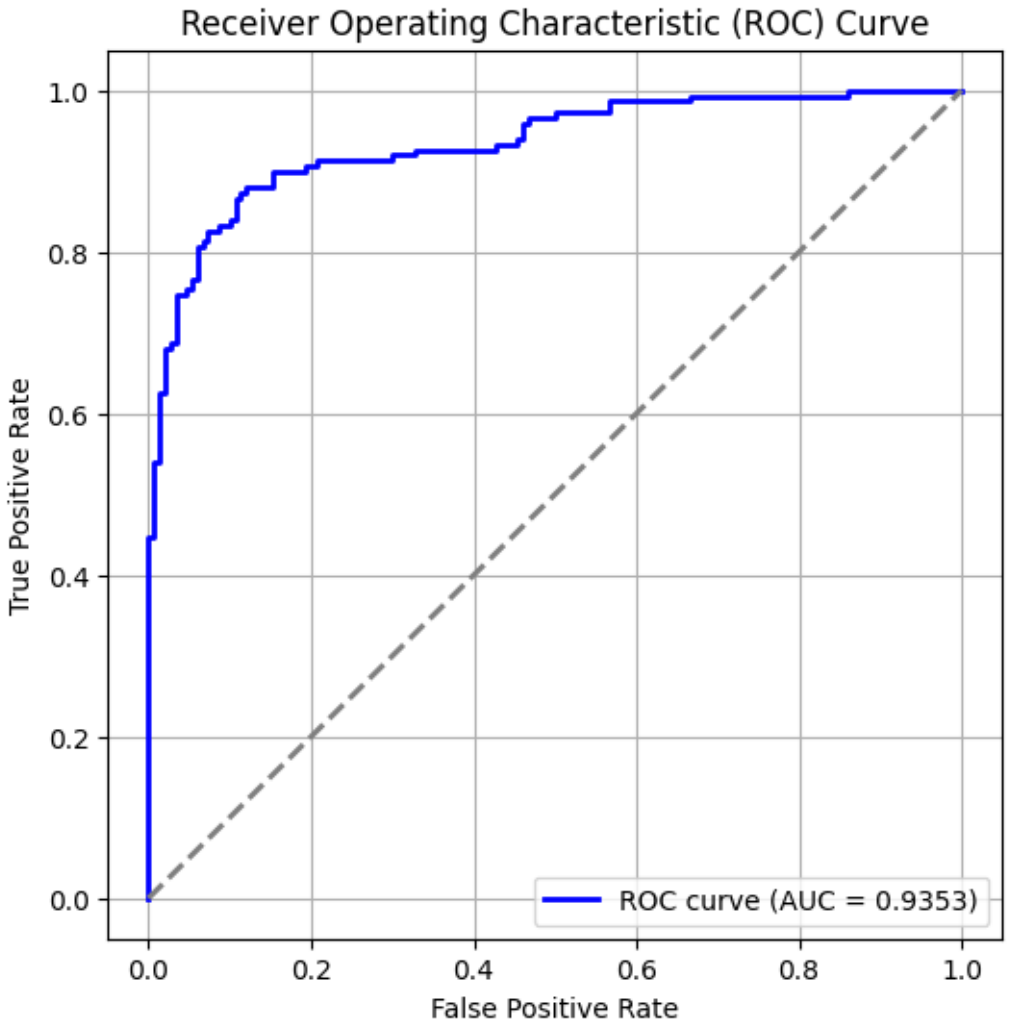
Recall (بازخوانی):

مقدار ۸۶٫۰۰٪ نشان می‌دهد که مدل توانسته است ۸۶٪ از ایمیل‌های Spam را به‌درستی شناسایی کند و تنها ۱۴٪ را از دست داده است.

F1-Score:

مقدار ۸۷٫۴۶٪ به‌عنوان میانگین هماهنگ Precision و Recall، تعادل مناسبی را در پیش‌بینی‌های مدل ارائه داده است.

نمودار ROC و مقدار AUC



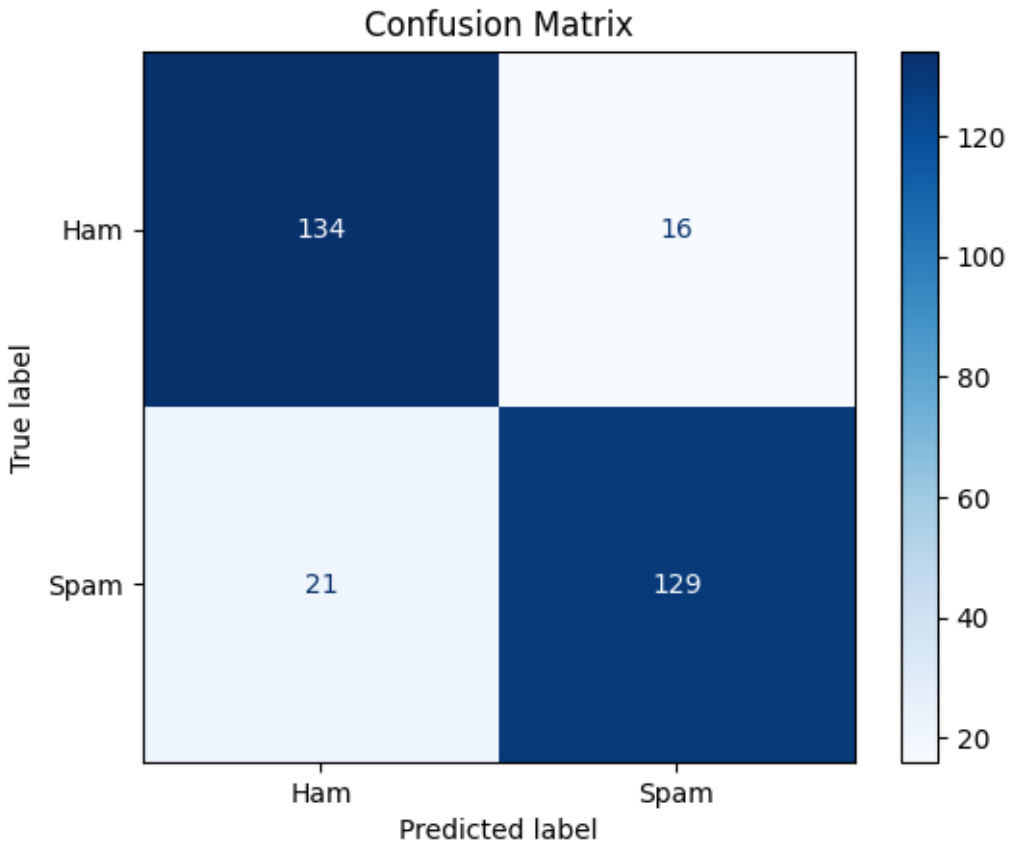
شکل ۶ نمودار **ROC** برای مدل **CNN**

نمودار ROC Curve نشان‌دهنده عملکرد تفکیک کلاس‌ها توسط مدل است. مقدار AUC = 0.9353 بیانگر این است که مدل در ۹۳٫۵۳٪ مواقع، پیش‌بینی‌های صحیحی ارائه کرده است

مقدار AUC بالاتر از ۰٫۹۰ نشان‌دهنده قدرت بالای مدل در تفکیک کلاس‌های Spam و Ham است.

منحنی ROC فاصله زیادی از خط مورب (تصادفی) دارد که بیانگر عملکرد قابل‌اطمینان مدل است.

ماتریس سردرگمی



شکل ۷ ماتریس سردرگمی برای مدل **CNN**

مدل ۱۳۴ ایمیل Ham را به‌درستی شناسایی کرده است.

تنها ۱۶ ایمیل Ham به اشتباه به‌عنوان Spam شناخته شده‌اند.

مدل ۱۲۹ ایمیل Spam را به درستی شناسایی کرده است.

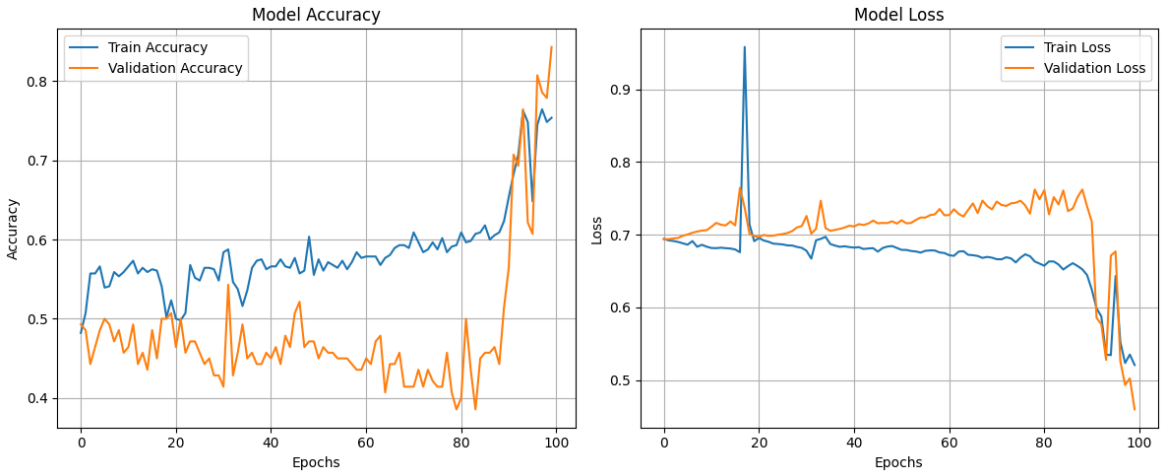
تنها ۲۱ ایمیل Spam به اشتباه Ham پیش‌بینی شده‌اند.

مدل LSTM

جدول **4** جدول معماری مدل **LSTM**

Layer (Type)	Output Shape	Param #
Embedding (Embedding)	(None, 150, 64)	384,000
LSTM (LSTM)	(None, 64)	33,024
Dense (Output)	(None, 1)	65

نمودار دقت و خطا داده های آموزشی و ارزیابی



شکل ۸ نمودار دقت و خطا در داده های آموزشی و ارزیابی برای مدل **LSTM**

نمودارهای بالا روند تغییر دقت (Accuracy) و خطا (Loss) را در طول ۱۰۰ ایپاک برای داده‌های آموزشی و اعتبارسنجی نشان می‌دهند.

ارزیابی مدل LSTM

معیار های ارزیابی مدل اینگونه شد

Accuracy: 0.7233

Precision: 0.6772

Recall: 0.8533

F1 Score: 0.7552

Accuracy (دقت کلی):

مقدار ۷۲٫۳۳٪ نشان‌دهنده توانایی متوسط مدل در پیش‌بینی صحیح کلاس‌ها است. این مقدار قابل قبول است، اما هنوز جای بهبود دارد.

Precision (دقت پیش‌بینی مثبت):

مقدار ۶۷٫۷۲٪ بیانگر این است که مدل در شناسایی موارد Spam عملکرد نسبتاً خوبی داشته است. اما درصد قابل توجهی از پیش‌بینی‌های مثبت، به اشتباه صورت گرفته‌اند.

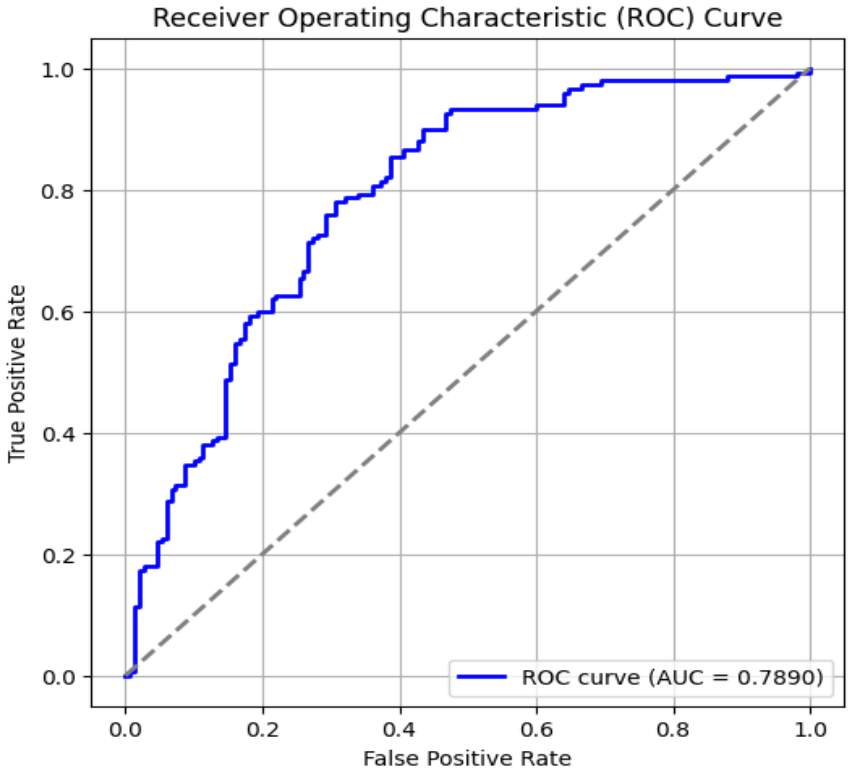
Recall (بازخوانی):

مقدار ۸۵٫۳۳٪ نشان می‌دهد که مدل بیشتر ایمیل‌های Spam را به درستی شناسایی کرده است. این معیار بالا نشان‌دهنده حساسیت مدل به شناسایی اسپم‌هاست.

F1-Score:

مقدار ۷۵٫۵۲٪ تعادل بین Precision و Recall را نشان می‌دهد. این مقدار تأیید می‌کند که مدل عملکرد متعادلی داشته اما نیاز به بهبود دقت دارد.

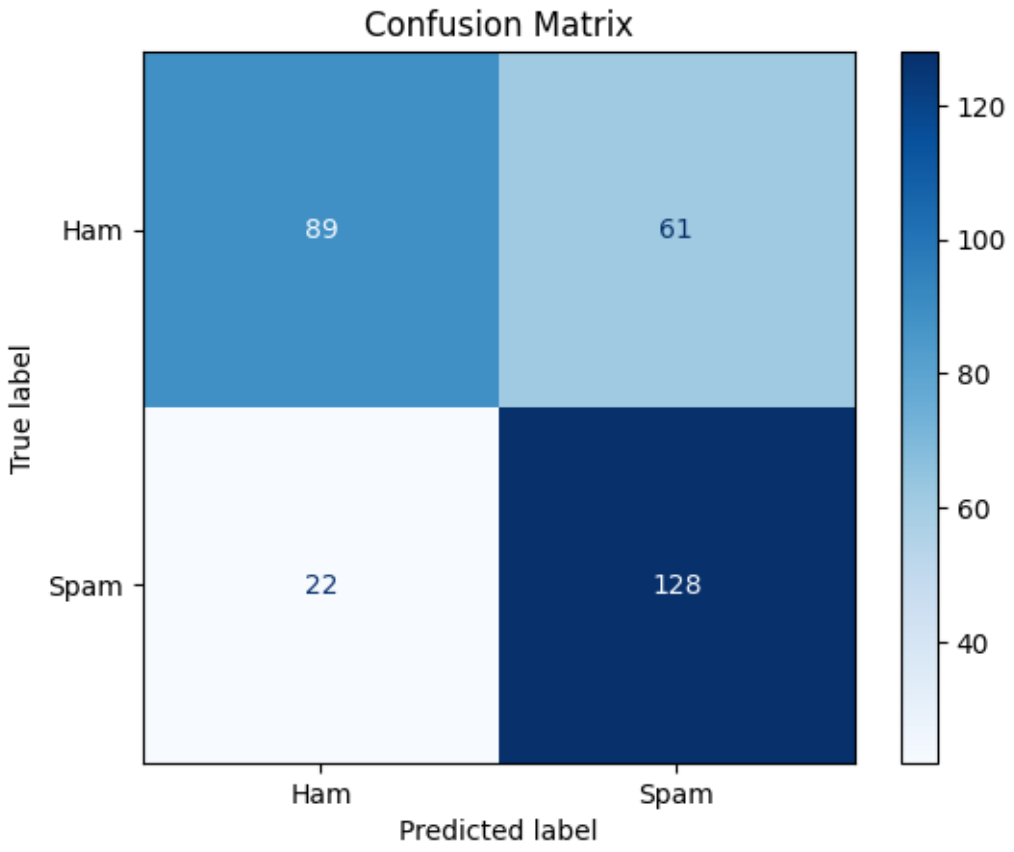
نمودار ROC و مقدار AUC



شکل ۹ نمودار **ROC** برای مدل **LSTM**

مقدار AUC زیر ۰٫۸۰ نشان می‌دهد که مدل هنوز به سطح اطمینان بالا برای جداسازی کلاس‌ها نرسیده است.

ماتریس سردرگمی



شکل ۱۰ ماتریس سردرگمی برای مدل **LSTM**

مدل ۸۹ ایمیل Ham را به‌درستی شناسایی کرده اما ۶۱ مورد را به اشتباه به‌عنوان Spam پیش‌بینی کرده است که نشان‌دهنده نرخ بالای False Positives است.

همچنین ۱۲۸ ایمیل Spam به‌درستی شناسایی شده‌اند، اما ۲۲ مورد به اشتباه Ham پیش‌بینی شده‌اند

سوال

نقاط قوت و ضعف هر یک از مدل‌ها CNN و LSTM چیست؟

مدل CNN (شبکه عصبی پیچشی):

نقاط قوت:

۱. **استخراج ویژگی‌های محلی:**

CNN در استخراج الگوهای محلی و روابط بین کلمات بسیار قدرتمند است و از فیلترهای پیچشی برای شناسایی الگوهای ناحیه‌ای استفاده می‌کند.

۲. **پردازش سریع‌تر:**

به دلیل ساختار لایه‌های موازی، CNNسرعت پردازش بالایی دارد و می‌تواند روی داده‌های بزرگ به صورت کارآمد کار کند.

۳. **مقاومت در برابر نویز:**

به دلیل عملیات حداکثر تجمع(MaxPooling) ، ویژگی‌های مهم حفظ شده و نویز کاهش پیدا می‌کند.

۴. **کاربرد مؤثر برای ورودی‌های ثابت طول:**

مناسب برای متون کوتاه و ورودی‌هایی که طول ثابت دارند.

نقاط ضعف:

۱. **عدم درک روابط ترتیبی:**

CNNبرای درک توالی‌ها و ارتباطات طولانی‌مدت در متن طراحی نشده است و تنها به ویژگی‌های مکانی توجه می‌کند.

۲. **محدودیت در پردازش داده‌های متغیر طول:**

اگر طول جملات متغیر باشد، نیاز به پیش‌پردازش بیشتری دارد تا به طول ثابت تبدیل شود.

۳. **عدم حافظه داخلی:**

CNNقابلیت نگهداری حافظه و مدیریت وابستگی‌های طولانی‌مدت را ندارد.

مدل LSTM (حافظه کوتاه-بلند مدت):

نقاط قوت:

۱. **مدیریت روابط ترتیبی:**

LSTMقادر به یادگیری وابستگی‌های زمانی و روابط طولانی‌مدت بین کلمات است که برای پردازش زبان طبیعی حیاتی است.

۲. **نگهداری اطلاعات گذشته:**

دارای سلول‌های حافظه است که اطلاعات مهم قبلی را حفظ کرده و در پیش‌بینی‌های فعلی استفاده می‌کند.

۳. **عملکرد قوی برای داده‌های متوالی:**

مناسب برای متون طولانی، مکالمات و ورودی‌هایی که به حافظه نیاز دارند.

نقاط ضعف:

۱. **زمان آموزش طولانی‌تر:**

به دلیل وجود وابستگی‌های ترتیبی، آموزش LSTM کندتر از CNN است.

۲. **نیاز به محاسبات بیشتر:**

پیچیدگی محاسباتی بالاتری دارد و نیاز به منابع سخت‌افزاری بیشتری مانند GPU دارد.

۳. **خطر بیش‌برازش:(Overfitting)**

به دلیل توانایی بالا در یادگیری، ممکن است بر روی داده‌های آموزشی بیش از حد تطبیق یابد و دقت روی داده‌های جدید کاهش یابد.

ادغام این دو مدل با چه هدفی انجام می‌شود؟

ادغام مدل‌های CNN و LSTM یک راهکار هوشمندانه برای بهره‌برداری از نقاط قوت هر دو مدل است. CNN به‌سرعت ویژگی‌های محلی را استخراج می‌کند و LSTM وابستگی‌های طولانی‌مدت را تحلیل می‌کند. این ترکیب نه‌تنها دقت را افزایش می‌دهد، بلکه مدل را قادر می‌سازد تا الگوهای پیچیده‌تر و متنوع‌تر را یاد بگیرد.

این مدل ترکیبی گزینه‌ای قدرتمند برای مسائل پیچیده زبان طبیعی و تحلیل داده‌های ترتیبی است.

۱-۵ ارزیابی

مقایسه نتایج ارزیابی سه مدل (CNN، CNN-LSTM، LSTM)

جدول **5** مقایسه نتیجه همه مدل‌ها

مدل	Accuracy	Precision	Recall	F1-Score	AUC
CNN	87.67%	88.97%	86.00%	87.46%	0.9353

CNN-LSTM	84.67%	82.50%	88.00%	85.16%	0.9067
LSTM	72.33%	67.72%	85.33%	75.52%	0.789

تحلیل و مقایسه مدل‌ها:

دقت (Accuracy):

مدل CNN با ۸۷٫۶۷٪ بالاترین دقت را دارد و مدل LSTM با ۷۲٫۳۳٪ پایین‌ترین دقت را نشان می‌دهد.

این اختلاف نشان می‌دهد که معماری پیچشی (CNN) در استخراج ویژگی‌های محلی کارآمدتر از LSTM بوده است.

دقت پیش‌بینی مثبت (Precision):

مدل CNN با ۸۸٫۹۷٪ بهترین عملکرد را در کاهش خطاهای مثبت کاذب داشته است.

مدل LSTM با ۶۷٫۷۲٪ کمترین دقت را نشان داده که نیاز به بهینه‌سازی دارد.

بازخوانی (Recall):

مدل LSTM با مقدار ۸۵٫۳۳٪ توانایی بهتری در شناسایی موارد Spam دارد، اما به دلیل Precision پایین، باعث افزایش خطای مثبت کاذب شده است.

مدل CNN-LSTM نیز با مقدار ۸۸٫۰۰٪ نشان داده که تعادل خوبی در شناسایی ایمیل‌های اسپم دارد.

F1-Score:

مقدار F1-Score در مدل CNN با ۸۷٫۴۶٪ بهترین تعادل بین Precision و Recall را ایجاد کرده است.

مدل LSTM با مقدار ۷۵٫۵۲٪ کمترین عملکرد را نشان می‌دهد و نیاز به بهبود دارد.

منحنی AUC:

مدل CNN با مقدار ۰٫۹۳۵۳ بهترین عملکرد را در جداسازی کلاس‌ها داشته است.

مدل LSTM با مقدار ۰٫۷۸۹۰ عملکردی ضعیف‌تر دارد که بیانگر احتمال بالای پیش‌بینی تصادفی در آن است.

نتیجه‌گیری کلی:

بهترین مدل:

مدل CNN در تمامی معیارها عملکرد بهتری داشته و به دلیل استخراج ویژگی‌های محلی قوی، تعادل خوبی بین معیارهای ارزیابی ارائه داده است.

مدل متوسط (CNN-LSTM):

مدل CNN-LSTM نیز عملکرد مناسبی دارد اما به دلیل پیچیدگی بالاتر، در برخی معیارها نسبت به CNN ضعیف‌تر عمل کرده است. با این حال، در شناسایی الگوهای طولانی عملکرد بهتری دارد و برای داده‌هایی با وابستگی ترتیبی پیچیده‌تر مناسب‌تر است.

مدل ضعیف‌تر (LSTM):

مدل LSTM با وجود توانایی در شناسایی روابط طولانی، به دلیل کمبود ویژگی‌های محلی (که CNN بهتر استخراج می‌کند) عملکرد ضعیف‌تری داشته است. این مدل برای بهبود نیاز به تنظیم‌های بیشتر و بهینه‌سازی پارامترها دارد

.

پاسخ ۲ – پیش‌بینی قیمت نفت خام

۲-۱. مقدمه

در این پروژه، ما به بررسی یکی از رایج‌ترین کاربردهای شبکه‌های حافظه‌دار، یعنی پیش‌بینی سری‌های زمانی، پرداخته‌ایم. سری‌های زمانی به دلیل ماهیت متوالی و وابستگی داده‌هایشان به مقادیر پیشین، به‌عنوان یکی از موضوعات پرکاربرد و چالش‌برانگیز در علوم داده شناخته می‌شوند. در این میان، پیش‌بینی قیمت نفت خام، به‌عنوان یکی از متغیرهای کلیدی در اقتصاد جهانی، از اهمیت ویژه‌ای برخوردار است. نوسانات این قیمت نه تنها تأثیر مستقیمی بر بازارهای انرژی دارد، بلکه بر سیاست‌های اقتصادی کشورهای مختلف نیز اثرگذار است.

ما در این مطالعه، تلاش کردیم با استفاده از چهار روش مختلف، قیمت نفت خام را پیش‌بینی کنیم. این روش‌ها شامل مدل‌های کلاسیک مانند ARIMA و SARIMA و همچنین مدل‌های یادگیری عمیق مانند LSTM، Bi-LSTM و GRU بودند. هدف اصلی ما این بود که دقت پیش‌بینی این روش‌ها را ارزیابی کنیم و درک بهتری از نقاط قوت و ضعف هرکدام به دست آوریم.

۲-۲. مجموعه دادگان و آماده‌سازی

-علاوه بر داده های nullموجود، ده درصد داده های ثبت شده را به صورت رندم حذف کنید.

در این مرحله از پروژه، ما داده‌های تاریخی مربوط به قیمت نفت خام را از منبع Yahoo Finance دانلود کردیم. این داده‌ها شامل اطلاعات روزانه قیمت نفت خام از سال ۲۰۱۰ تا تاریخ کنونی بودند و ویژگی اصلی که برای پیش‌بینی انتخاب کردیم، ستون Adj Close بود. ستون Adj Close نشان‌دهنده قیمت تعدیل‌شده نفت خام در پایان هر روز است و به‌عنوان یکی از مهم‌ترین شاخص‌ها برای تحلیل‌های مالی و اقتصادی مورد استفاده قرار می‌گیرد.

پس از دریافت داده‌ها، اولین گام ما بررسی کامل مجموعه داده‌ها برای شناسایی تاریخ‌های گمشده بود. در بسیاری از روزها هیچ داده‌ای ثبت نشده بود که به‌عنوان مقادیر Null شناسایی شدند. برای این کار، از ابزارهای تحلیل داده استفاده کردیم و تعداد کل روزهایی که داده‌ها در آن‌ها گم شده بود را مشخص کردیم.

نتایج نشان داد که از مجموع ۵۴۶۷ روز موجود در بازه زمانی مورد بررسی، ۱۷۰۳ روز فاقد داده بودند. این نشان می‌دهد که حدود ۳۱ درصد از داده‌های اصلی ما گم شده‌اند.

برای ایجاد شرایط واقعی‌تر و بررسی عملکرد مدل‌ها در مواجهه با داده‌های ناقص، تصمیم گرفتیم که به‌طور تصادفی ۱۰ درصد دیگر از داده‌های موجود را حذف کنیم. برای این کار، ابتدا تاریخ‌هایی را که دارای داده‌های معتبر بودند شناسایی کردیم. سپس با استفاده از یک الگوریتم تصادفی، ۱۰ درصد از این تاریخ‌ها را انتخاب کرده و مقادیر Adj Close مربوط به آن‌ها را به‌طور دستی حذف کردیم. این فرآیند شبیه‌سازی سناریوهایی بود که ممکن است در دنیای واقعی رخ دهند، مانند گم شدن داده‌ها به دلیل مشکلات فنی یا خطاهای ثبت اطلاعات.

بعد از اعمال این تغییرات، تعداد روزهای گمشده در مجموعه داده به ۲۲۴۹ روز افزایش یافت. این بدان معناست که اکنون ۴۱ درصد از کل داده‌های موجود در مجموعه داده ما ناقص هستند. چنین شرایطی به ما این امکان را می‌دهد که مدل‌های پیش‌بینی را در مواجهه با داده‌های ناقص ارزیابی کنیم و روش‌های مناسبی برای مدیریت این داده‌ها ارائه دهیم.

در پایان این بخش، داده‌های نهایی ما شامل دو نوع داده گمشده بود: داده‌های گمشده اصلی که از منبع Yahoo Finance دریافت شده بودند و داده‌هایی که به‌طور تصادفی حذف کردیم. این مجموعه داده با چالش‌های واقعی روبرو بود و مرحله مهمی برای آماده‌سازی داده‌ها جهت آموزش مدل‌های پیش‌بینی بود. در ادامه، نحوه پر کردن داده‌های گمشده و آماده‌سازی مجموعه داده برای مدل‌سازی تشریح خواهد شد.

سپس، روش هایی برای جایگزینی داده های ناموجود ارائه دهید و داده ها را تکمیل کنید.

```
# Fill missing values
filled_data = data[['Adj Close']].copy()
filled_data['Adj Close'] = filled_data['Adj Close'].interpolate(method='linear')
filled_data['Adj Close'] = filled_data['Adj Close'].fillna(method='bfill').fillna(method='ffill')
```

شکل ۱۱ نحوه پر کردن داده های از دست رفته

برای تکمیل داده‌های گمشده در این پروژه، چندین روش متفاوت را مورد استفاده قرار دادیم تا مطمئن شویم داده‌ها به شکل مناسبی تکمیل شده و آماده مدل‌سازی هستند. این روش‌ها شامل میان‌یابی خطی، جایگزینی به کمک روش‌های Forward Fill و Backward Fill، و در نهایت ترکیبی از این دو روش بودند. در ادامه هر یک از این روش‌ها و دلایل استفاده از آن‌ها به تفصیل توضیح داده شده است.

ابتدا، روش میان‌یابی خطی را اعمال کردیم. این روش یکی از ساده‌ترین و در عین حال مؤثرترین روش‌های جایگزینی داده‌های گمشده است. در میان‌یابی خطی، مقدار گمشده بر اساس مقادیر قبل و بعد از آن تخمین زده می‌شود. این روش فرض می‌کند که تغییرات داده‌ها در بازه زمانی گمشده به صورت خطی است و مقدار میانی به‌صورت خطی بین مقادیر قبل و بعد محاسبه می‌شود. برای مثال، اگر مقدار روز دوم و چهارم مشخص باشد ولی مقدار روز سوم گم شده باشد، این روش مقدار روز سوم را به‌صورت میانگین خطی بین این دو مقدار تخمین می‌زند. دلیل انتخاب این روش این بود که قیمت نفت خام معمولاً روندهای تدریجی دارد و تغییرات آن در بازه‌های زمانی کوتاه به شکل ناگهانی نیست، بنابراین این روش می‌تواند تخمین مناسبی ارائه دهد.

سپس، برای تکمیل داده‌هایی که در ابتدای مجموعه یا انتهای آن گمشده بودند و امکان میان‌یابی خطی نداشتند، از روش‌های Forward Fill و Backward Fill استفاده کردیم. در روش Forward Fill، مقدار گمشده با آخرین مقدار موجود پیش از آن جایگزین می‌شود. به همین ترتیب، در روش Backward Fill، مقدار گمشده با اولین مقدار موجود پس از آن پر می‌شود. این روش‌ها زمانی مفید هستند که فرض کنیم قیمت نفت خام در بازه‌های زمانی کوتاه‌مدت نسبتاً پایدار است و مقدار فعلی به مقادیر قبلی یا بعدی نزدیک خواهد بود. دلیل استفاده از این روش‌ها این بود که در ابتدای مجموعه داده و انتهای آن، ممکن بود هیچ مقدار معتبری برای انجام میان‌یابی خطی وجود نداشته باشد. به همین دلیل، این روش‌ها به‌عنوان راه‌حل مکمل به میان‌یابی خطی اضافه شدند.

برای اطمینان از تکمیل کامل داده‌ها، ترکیبی از روش‌های ذکر شده را به کار بردیم. ابتدا از میان‌یابی خطی برای پر کردن داده‌های گمشده در میان بازه‌های معتبر استفاده شد. سپس، برای داده‌هایی که در ابتدا یا انتهای مجموعه گمشده بودند و امکان میان‌یابی نداشتند، از Forward Fill و Backward Fill استفاده کردیم. این ترکیب به ما اجازه داد که تمامی داده‌های گمشده را به‌طور کامل تکمیل کنیم.

دلیل انتخاب این روش‌ها ساده بودن آن‌ها و کارایی بالا در مواجهه با داده‌هایی است که تغییرات تدریجی دارند. همچنین این روش‌ها نیازی به پیچیدگی‌های محاسباتی نداشته و به‌راحتی بر روی مجموعه داده‌های بزرگ قابل اعمال هستند. پس از اعمال این مراحل، تمامی مقادیر گمشده تکمیل شدند و داده‌ها برای مراحل بعدی آماده‌سازی شدند.

طبق نسبت موجود در مقاله داده هارا به دو دسته ی آموزشی و آزمایشی تقسیم کرده و نرمال کنید

در این بخش از پروژه، ما داده‌ها را طبق نسبت ارائه‌شده در مقاله به دو دسته‌ی آموزشی و آزمایشی تقسیم کردیم و سپس داده‌ها را برای آموزش مدل‌ها نرمال‌سازی کردیم. این مرحله یکی از گام‌های اساسی در آماده‌سازی داده‌ها برای مدل‌سازی بود، زیرا تقسیم مناسب داده‌ها و نرمال‌سازی صحیح آن‌ها می‌تواند تأثیر مستقیم بر عملکرد و دقت مدل‌ها داشته باشد.

ابتدا مجموعه داده‌های تکمیل‌شده که شامل مقادیر پر شده برای داده‌های گمشده بود، آماده شد. سپس طبق روش مقاله، داده‌ها به دو بخش تقسیم شدند. ۷۰ درصد داده‌ها به‌عنوان داده‌های آموزشی و ۳۰ درصد باقی‌مانده به‌عنوان داده‌های آزمایشی در نظر گرفته شدند. این نسبت به‌طور خاص برای مسائل سری زمانی بسیار مناسب است. دلیل این انتخاب این است که در مسائل پیش‌بینی سری زمانی، مدل‌ها نیاز دارند که تعداد کافی داده برای یادگیری الگوها داشته باشند، و در عین حال مجموعه داده‌ای نیز برای ارزیابی عملکرد باقی بماند. این نسبت، تعادلی میان داده‌های مورد نیاز برای آموزش و آزمایش ایجاد می‌کند.

برای انجام این کار، داده‌ها به ترتیب زمانی تقسیم شدند. به این معنا که داده‌های قدیمی‌تر در مجموعه‌ی آموزشی قرار گرفتند و داده‌های جدیدتر به مجموعه‌ی آزمایشی اختصاص داده شدند. این رویکرد، سازگاری با ماهیت سری زمانی دارد، چرا که در مسائل سری زمانی، داده‌های جدیدتر معمولاً به‌عنوان مقادیر ناشناخته‌ای که قرار است پیش‌بینی شوند، استفاده می‌شوند و نباید در فرآیند آموزش مدل لحاظ شوند.

پس از تقسیم‌بندی داده‌ها، مرحله‌ی نرمال‌سازی آغاز شد. نرمال‌سازی داده‌ها به این دلیل انجام می‌شود که مدل‌های مبتنی بر یادگیری عمیق مانند LSTM، Bi-LSTM و GRU معمولاً نسبت به مقیاس داده‌ها حساس هستند. اگر داده‌ها در مقیاس‌های مختلف باشند، ممکن است مدل‌ها به درستی نتوانند الگوهای موجود در داده‌ها را یاد بگیرند یا الگوریتم‌های بهینه‌سازی به‌کندی همگرا شوند. برای جلوگیری از این مشکل، تمامی مقادیر داده‌ها به بازه‌ای مشخص، معمولاً بین ۰ و ۱، نگاشت شدند.

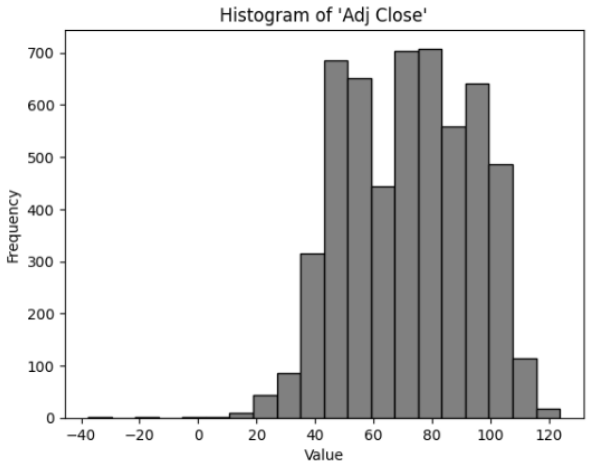
برای نرمال‌سازی، از ابزار MinMaxScaler که یکی از ابزارهای موجود در کتابخانه Scikit-learn است، استفاده کردیم. این ابزار داده‌ها را بر اساس فرمول مشخصی به بازه‌ی بین ۰ و ۱ تبدیل می‌کند. در این روش، هر مقدار با استفاده از کمینه و بیشینه داده‌ها مقیاس‌بندی می‌شود، به‌گونه‌ای که کوچک‌ترین مقدار برابر با ۰ و بزرگ‌ترین مقدار برابر با ۱ خواهد بود.

ابتدا داده‌های آموزشی را با استفاده از این ابزار نرمال‌سازی کردیم. سپس، برای داده‌های آزمایشی نیز از همان مقیاس استفاده شد که بر اساس داده‌های آموزشی به دست آمده بود. این نکته بسیار مهم است، زیرا داده‌های آزمایشی باید با همان مقیاس داده‌های آموزشی مقایسه شوند تا مدل بتواند نتایج قابل اعتمادی ارائه دهد.

این فرآیند نرمال‌سازی باعث شد که تمامی مقادیر داده‌ها در بازه‌ی یکسانی قرار بگیرند. این کار به مدل‌ها کمک می‌کند که بهتر و سریع‌تر الگوها را یاد بگیرند و از تأثیر منفی مقادیر بزرگ یا کوچک بر عملکرد مدل جلوگیری می‌کند.

مشابه شکل ۶داخل مقاله، هیستوگرام توزیع قیمت را نمایش دهید.

```
[9]:
# Plot histogram of 'Adj Close'
import matplotlib.pyplot as plt
plt.hist(filled_data['Adj Close'], bins=20, color='gray', edgecolor='black')
plt.title("Histogram of 'Adj Close'")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```



شکل ۱۲هیستوگرام توزیع قیمت

در این بخش از پروژه، هیستوگرام داده‌های موجود در ستون Adj Close رسم شد. هدف اصلی از این کار، بررسی توزیع داده‌ها و شناسایی الگوهای کلیدی در قیمت نفت خام بود. هیستوگرام ابزاری بسیار مؤثر برای نمایش بصری توزیع داده‌ها است که می‌تواند به ما کمک کند تا درک بهتری از رفتار داده‌ها و ویژگی‌های آماری آن‌ها به دست آوریم.

برای این کار، از داده‌های تکمیل‌شده استفاده کردیم، یعنی داده‌هایی که پس از پر کردن مقادیر گمشده و پیش‌پردازش آماده شده بودند. هیستوگرام با استفاده از کتابخانه Matplotlib در پایتون رسم شد. این کتابخانه یکی از ابزارهای قدرتمند و پرکاربرد برای مصورسازی داده‌ها است که امکان تنظیم ویژگی‌های مختلف نمودارها را در اختیار ما قرار می‌دهد.

در رسم هیستوگرام، تعداد دسته‌ها برابر با ۲۰ انتخاب شد. این تعداد به ما اجازه داد تا داده‌ها را به طور دقیق در بازه‌های مختلف تقسیم‌بندی کنیم و تغییرات در توزیع داده‌ها را بهتر مشاهده کنیم. رنگ میله‌های

هیستوگرام خاکستری و لبه‌های آن مشکی تعیین شد تا نمایش داده‌ها به‌صورت شفاف و خوانا باشد. محور افقی هیستوگرام نمایانگر مقادیر Adj Closeو محور عمودی نشان‌دهنده‌ی تعداد دفعاتی است که هر مقدار در داده‌ها ظاهر شده است.

نتیجه‌ی رسم هیستوگرام نشان داد که داده‌ها در بازه‌ای مشخص متمرکز شده‌اند. بیشترین تمرکز داده‌ها در محدوده‌ی قیمت‌های بین ۴۰ تا ۱۰۰ بود. این الگو نشان‌دهنده‌ی این است که قیمت نفت خام در اکثر روزهای بررسی‌شده در این بازه قرار داشته است. همچنین، توزیع داده‌ها به‌صورت نسبتاً متقارن به نظر می‌رسد و نوسانات شدید یا مقادیر پرت در این بازه کمتر مشاهده می‌شود.

این تحلیل به ما کمک کرد تا پیش از ورود به مراحل پیش‌بینی، از توزیع داده‌ها آگاهی پیدا کنیم. با استفاده از این ابزار، توانستیم مشخص کنیم که داده‌ها در اکثر موارد چگونه رفتار می‌کنند و آیا الگوهای خاصی در آن‌ها وجود دارد یا خیر. در صورتی که داده‌ها دارای انحراف‌های غیرعادی یا رفتارهای نامتعارف بودند، این موارد در هیستوگرام به وضوح نمایان می‌شد و می‌توانستیم آن‌ها را مدیریت کنیم.

۲-۳. پیاده‌سازی مدل‌ها

برای پیش‌بینی سری زمانی در این پروژه، از سه مدل شبکه عصبی پیشرفته شامل LSTM، Bi-LSTM و GRU استفاده شد. این مدل‌ها به دلیل قابلیت خاصی که در یادگیری الگوهای زمانی و روابط طولانی‌مدت دارند، در مسائل مرتبط با سری زمانی بسیار پرکاربرد هستند. هدف ما این بود که با استفاده از این سه مدل و تنظیم دقیق هایپرپارامترها، به پیش‌بینی دقیق قیمت نفت خام بپردازیم. در ادامه، روش کار و فرآیند آموزش این مدل‌ها با جزئیات کامل توضیح داده می‌شود.

ابتدا به سراغ مدل LSTM رفتیم. این مدل که مخفف Long Short-Term Memory است، یک نوع پیشرفته از شبکه‌های بازگشتی است که برای رفع مشکل از بین رفتن گرادیان‌ها در طول فرآیند یادگیری طراحی شده است. LSTM با استفاده از سلول حافظه و دروازه‌هایی برای کنترل اطلاعات ورودی، خروجی و فراموشی، قادر است اطلاعات کلیدی را برای مدت طولانی حفظ کند و اطلاعات غیرضروری را حذف کند. در پروژه ما، مدل LSTM با استفاده از ۵۱۲ واحد مخفی طراحی شد. این تعداد واحدها، طبق تنظیمات ارائه‌شده در مقاله انتخاب شدند تا مدل بتواند الگوهای پیچیده در داده‌ها را به‌خوبی یاد بگیرد. برای لایه خروجی، یک لایه Dense با یک نرون در نظر گرفته شد که وظیفه تولید پیش‌بینی نهایی را داشت. تابع خطا برای آموزش مدل، میانگین مربعات خطا نتخاب شد، چرا که این تابع خطا برای مسائل رگرسیون و سری زمانی بسیار مناسب است. همچنین، از بهینه‌ساز Adam با نرخ یادگیری ۰,۰۰۱ استفاده کردیم که به دلیل سرعت همگرایی بالا و کارایی مناسب در مسائل پیچیده انتخاب شد. مدل در ۵۰ دوره

آموزشی و با اندازه دسته ۱۰۰ آموزش داده شد. این تنظیمات، تعادل مناسبی میان دقت و زمان آموزش ایجاد کردند.

در ادامه، مدل Bi-LSTM آموزش داده شد. این مدل نسخه‌ای پیشرفته‌تر از LSTM است که داده‌ها را در هر دو جهت زمانی (از گذشته به آینده و از آینده به گذشته) پردازش می‌کند. این ویژگی باعث می‌شود که مدل بتواند الگوهای زمانی پیچیده‌تر را شناسایی کند و اطلاعات بیشتری را از داده‌ها استخراج کند. ساختار Bi-LSTM نیز مشابه LSTM بود، با این تفاوت که تعداد واحدهای مخفی در این مدل به ۱۰۲۴ افزایش یافت. سایر تنظیمات از جمله تابع خطا، بهینه‌ساز، تعداد دوره‌ها و اندازه دسته‌ها همانند LSTM بودند. این مدل به دلیل استفاده از دو جهت زمانی، توانایی بالاتری در پیش‌بینی الگوهای پنهان سری زمانی دارد.

مدل سوم مورد استفاده، GRU بود که مخفف Gated Recurrent Unit است. GRU نسخه ساده‌تر و سریع‌تر از LSTM است که با حذف برخی از دروازه‌ها، ساختار ساده‌تری دارد. این مدل، به‌ویژه در مواردی که داده‌ها طولانی و پیچیده هستند، کارایی بالاتری ارائه می‌دهد. ساختار GRU در پروژه ما شامل ۵۱۲ واحد مخفی بود و سایر تنظیمات مشابه مدل LSTM بودند. سادگی GRU باعث شد که این مدل بتواند با سرعت بیشتری آموزش ببیند و نتایج قابل قبولی ارائه دهد.

برای آموزش این مدل‌ها، ابتدا داده‌ها آماده‌سازی شدند. داده‌های سری زمانی به دو بخش آموزشی و آزمایشی تقسیم شدند، به‌گونه‌ای که ۷۰ درصد داده‌ها برای آموزش و ۳۰ درصد باقی‌مانده برای ارزیابی مدل‌ها مورد استفاده قرار گرفتند. سپس، داده‌ها با استفاده از MinMaxScaler نرمال‌سازی شدند تا تمامی مقادیر داده‌ها در بازه‌ای مشخص (معمولاً بین ۰ و ۱) قرار گیرند. این کار باعث شد که الگوریتم‌های بهینه‌سازی در مدل‌های عصبی بتوانند با سرعت و دقت بیشتری یادگیری را انجام دهند.

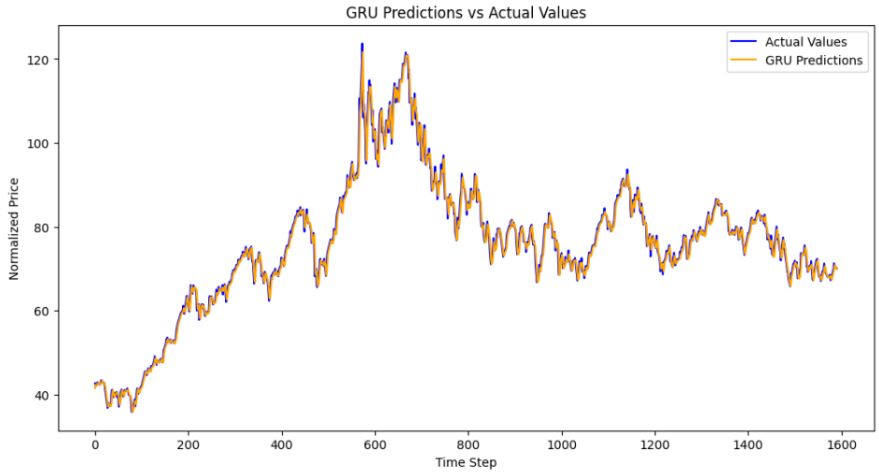
برای تبدیل داده‌ها به فرمت قابل استفاده در مدل‌های شبکه عصبی، از تکنیکی به نام Sliding Window استفاده شد. در این تکنیک، برای پیش‌بینی مقدار روز بعد، از ۵۰ مقدار قبلی به‌عنوان ورودی استفاده کردیم. این بازه زمانی (Window Size) نیز طبق پیشنهادات مقاله انتخاب شد. داده‌های تبدیل‌شده سپس به شکل توالی‌هایی مناسب برای ورودی مدل‌های LSTM، Bi-LSTM و GRU در آمدند.

هر مدل به‌صورت جداگانه و با تنظیمات ارائه‌شده در مقاله آموزش داده شد. فرآیند آموزش به گونه‌ای بود که در هر دوره آموزشی، مدل‌ها داده‌های آموزشی را تحلیل و خروجی پیش‌بینی‌شده را با مقادیر واقعی مقایسه می‌کردند. این مقایسه با استفاده از تابع خطای میانگین مربعات خطا انجام شد و مدل‌ها با به‌روزرسانی وزن‌ها به کمک بهینه‌ساز Adam، تلاش کردند که خطای پیش‌بینی را به حداقل برسانند.

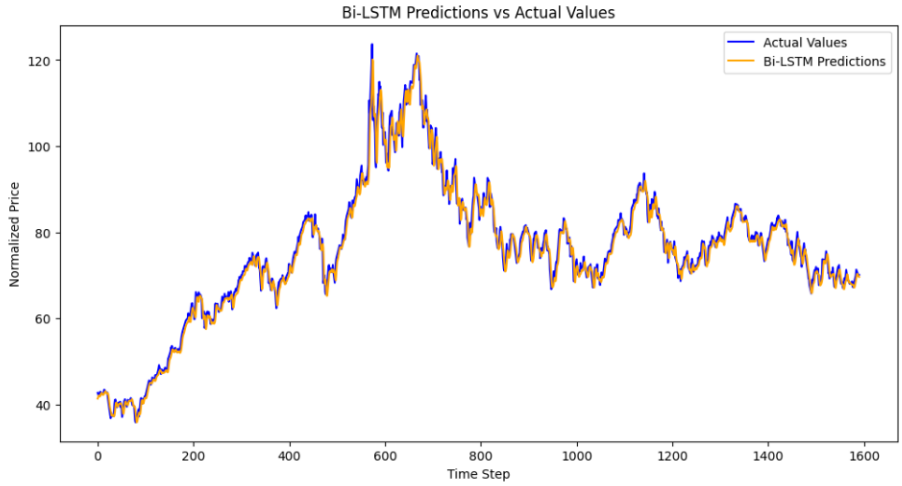
پس از اتمام فرآیند آموزش، مدل‌ها بر روی داده‌های آزمایشی ارزیابی شدند. خروجی مدل‌ها به‌صورت مقادیر نرمال‌سازی‌شده تولید شدند که با استفاده از مقیاس اصلی داده‌ها، به مقادیر واقعی بازگردانده شدند. این خروجی‌ها سپس با مقادیر واقعی مقایسه شدند و عملکرد هر مدل با استفاده از معیارهایی مانند MAE، RMSE، R2 و MAPE اندازه‌گیری شد.

تمامی مراحل فوق با دقت مطابق روش‌های ارائه‌شده در مقاله انجام شد. تنظیمات مربوط به هایپرپارامترها و ساختار مدل‌ها از جدول ۴ مقاله استخراج شدند.

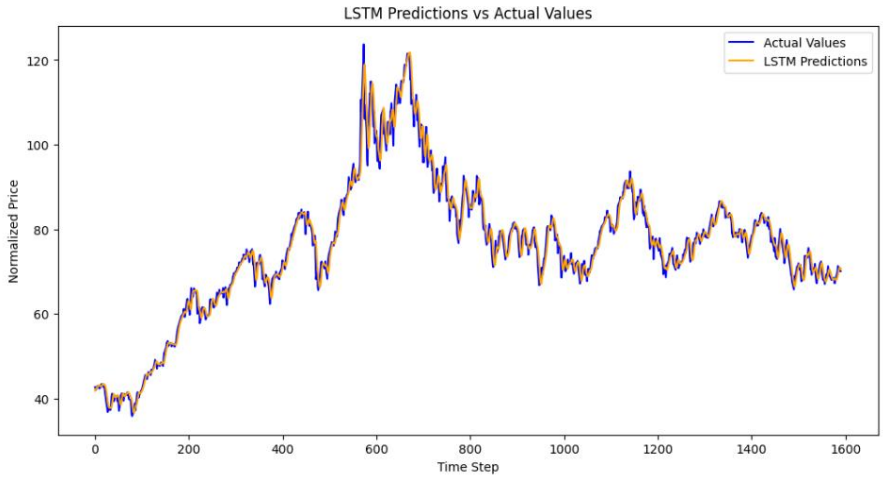
برای هر سه مدل داده شده، نتایج پیشبینی شده را همراه مقادیر واقعی نمایش دهید



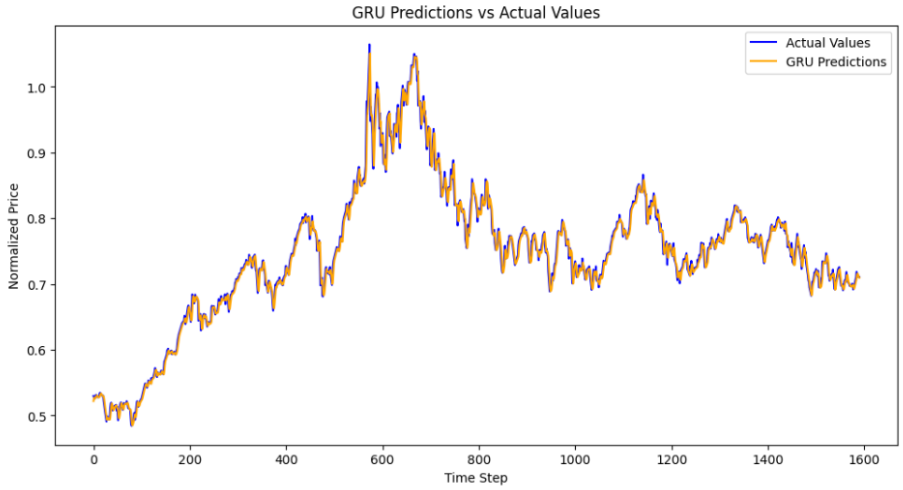
شکل ۱۳ مقایسه پیش‌بینی های **GRU** با مقادیر واقعی



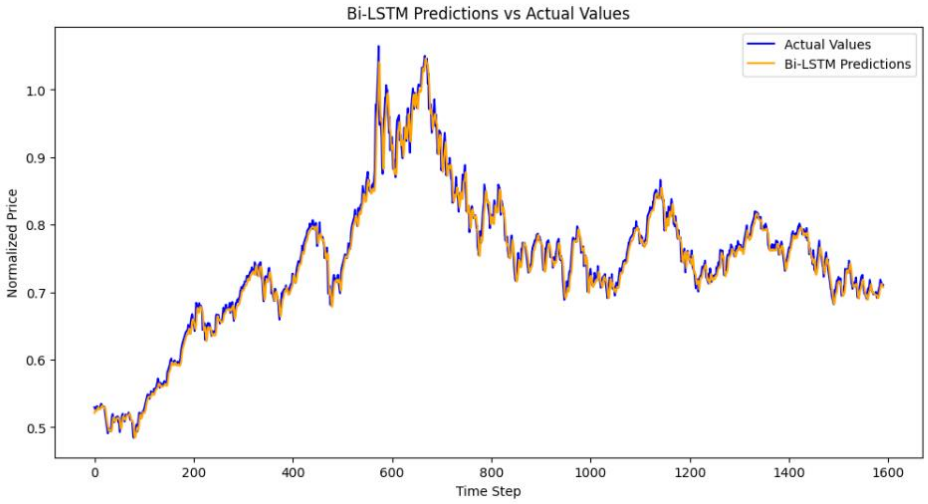
شکل ۱۴مقایسه پیش‌بینی‌های **Bi-LSTM** با مقادیر واقعی



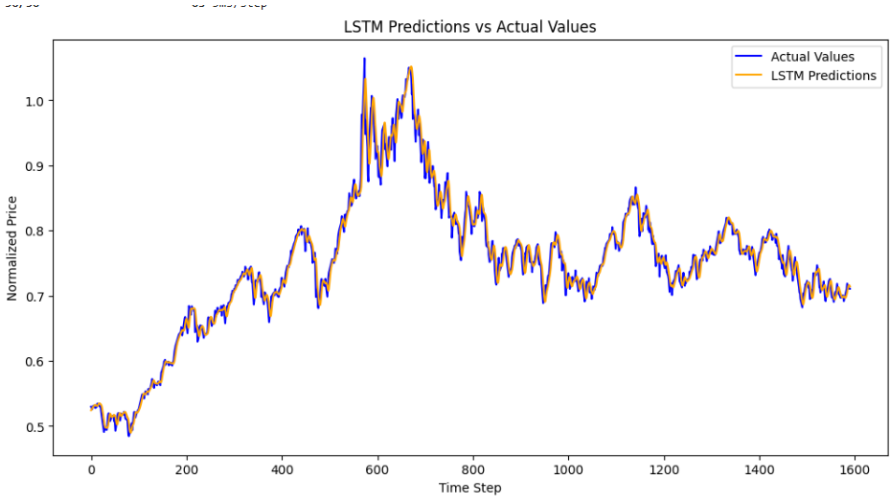
شکل ۱۵ مقایسه پیش‌بینی‌های **LSTM** با مقادیر واقعی



شکل ۱۶ مقایسه پیش‌بینی‌های **GRU** با مقادیر اسکیل شده



شکل ۱۷ مقایسه پیش‌بینی‌های **Bi-LSTM** با مقادیر اسکیل شده



شکل ۱۸مقایسه مقادیر پیش‌بینی شده **LSTM** با مقادیر واقعی اسکیل شده

برای بررسی عملکرد مدل‌های LSTM، Bi-LSTM و GRU در پیش‌بینی سری زمانی قیمت نفت خام، نتایج پیش‌بینی‌های این مدل‌ها را با مقادیر واقعی مقایسه کردیم. این مقایسه شامل دو مرحله اصلی بود: نخست، نمایش نتایج بر اساس داده‌های نرمال‌سازی شده و سپس، تبدیل خروجی‌ها به مقادیر واقعی و تحلیل آن‌ها. این دو رویکرد به ما امکان داد تا به‌صورت دقیق‌تر عملکرد مدل‌ها را از جنبه‌های مختلف بررسی کنیم.

برای آماده‌سازی داده‌ها جهت آموزش مدل‌های حافظه‌دار مانندLSTM ، Bi-LSTMوGRU ، ابتدا داده‌ها به دنباله‌هایی (Sequences) با طول ثابت تقسیم شدند. این فرآیند که به عنوان ایجاد توالی شناخته می‌شود، به مدل اجازه می‌دهد تا الگوها و وابستگی‌های زمانی موجود در داده‌ها را یاد بگیرد. به این منظور، یک تابع برای ساخت دنباله‌ها تعریف شد که به طور مداوم از یک بازه مشخص از داده‌ها (به عنوان ورودی) استفاده می‌کرد و مقدار بعدی (به عنوان هدف) را به‌عنوان خروجی ثبت می‌کرد. در این پروژه، طول بازه (window size)برابر با ۵۰ در نظر گرفته شد، به این معنا که هر توالی شامل ۵۰ مقدار قبلی است و مقدار ۵۱ام به عنوان هدف آن توالی تعریف می‌شود. این ساختار به مدل کمک می‌کند تا وابستگی‌های طولانی‌مدت و الگوهای موجود در داده‌های سری زمانی را شناسایی کند. پس از ایجاد این توالی‌ها، داده‌ها به فرمت سه‌بعدی تبدیل شدند تا قابل استفاده در مدل‌های حافظه‌دار شوند، به طوری که هر توالی به شکل (تعداد نمونه‌ها، طول توالی، تعداد ویژگی‌ها) تعریف شد. این مرحله نقش مهمی در آموزش موفق مدل‌ها و استخراج اطلاعات زمانی از داده‌ها داشت.

ابتدا، مدل‌های آموزش‌دیده برای پیش‌بینی مقادیر سری زمانی در داده‌های آزمایشی به کار گرفته شدند. این پیش‌بینی‌ها ابتدا بر اساس مقادیر نرمال‌سازی‌شده‌ای که در فرآیند آموزش استفاده شده بودند، ارائه شدند. در این حالت، مقادیر واقعی نیز در همان محدوده مقیاس نرمال‌سازی قرار داشتند و نمودارهایی تولید شد که مقادیر پیش‌بینی‌شده و واقعی را در یک بازه نرمال‌شده نمایش می‌دادند. این نمودارها به ما کمک کردند تا الگوهای کلی پیش‌بینی‌ها و دقت نسبی هر مدل را در پیش‌بینی سری‌های زمانی مشاهده کنیم.

مدل LSTM توانست به‌طور قابل توجهی الگوهای سری زمانی را شناسایی کرده و پیش‌بینی‌هایی بسیار نزدیک به مقادیر واقعی ارائه دهد. نمودار مقایسه‌ای LSTM نشان داد که این مدل توانسته است تغییرات قیمت را به‌خوبی دنبال کند و نوسانات را شبیه‌سازی کند.

Bi-LSTM نیز عملکرد بسیار مشابهی داشت و در برخی بخش‌ها حتی توانایی بیشتری در پیش‌بینی دقیق‌تر مقادیر از خود نشان داد. دلیل این عملکرد بهتر، توانایی این مدل در پردازش داده‌ها از دو جهت زمانی است که به آن امکان می‌دهد اطلاعات بیشتری از سری زمانی استخراج کند.

مدل GRU نیز نتایج قابل قبولی ارائه داد و توانست الگوهای زمانی را به‌خوبی شناسایی کند. با این حال، در برخی بازه‌ها دقت این مدل کمی کمتر از LSTM و Bi-LSTM بود، که می‌تواند به دلیل ساختار ساده‌تر و تعداد پارامترهای کمتر آن باشد.

در گام دوم، خروجی پیش‌بینی‌های مدل‌ها به مقیاس اصلی داده‌ها بازگردانده شد. این مرحله بسیار مهم بود، زیرا تحلیل مقادیر واقعی پیش‌بینی‌شده امکان ارزیابی دقیق‌تر عملکرد مدل‌ها را فراهم کرد. نمودارهای مقایسه‌ای در این مرحله نشان دادند که مدل‌های LSTM و Bi-LSTM در شناسایی نوسانات و روندهای کلی قیمت عملکرد بسیار خوبی داشتند و پیش‌بینی‌های آن‌ها با مقادیر واقعی هم‌خوانی بالایی داشت. GRU نیز با وجود تفاوت‌های جزئی، توانست نتایجی قابل قبول ارائه دهد.

به طور کلی، مقایسه علمی این مدل‌ها نشان داد که:

مدل Bi-LSTM به دلیل توانایی پردازش داده‌ها از دو جهت زمانی، در برخی بخش‌ها دقت بیشتری نسبت به LSTM داشت، اما این تفاوت به‌اندازه‌ای نبود که بتوان گفت عملکرد LSTM ضعیف‌تر است.

مدل GRU با ساختار ساده‌تر خود توانست نتایجی نزدیک به دو مدل دیگر ارائه دهد، اما در برخی بازه‌ها به‌ویژه در نوسانات شدید، کمی ضعیف‌تر عمل کرد.

از لحاظ کمی، معیارهای عملکرد مانند MAE، RMSE، R2 و MAPE برای هر سه مدل محاسبه و مقایسه شدند. این معیارها نشان دادند که تمامی مدل‌ها توانسته‌اند با خطای کم و دقت بالا پیش‌بینی‌های خود را انجام دهند. LSTM و Bi-LSTM در این معیارها عملکرد بهتری داشتند، در حالی که GRU در معیارهای خطا کمی تفاوت داشت اما همچنان قابل قبول بود.

ابتدا به طور مختصر در مورد معیار های داخل مقاله، R-Squared، RMSE،MAEو MAPE توضیح دهید. سپس مقادیر را گزارش کرده و نتایج را تحلیل و مقایسه کنید.

برای ارزیابی عملکرد مدل‌های پیش‌بینی، از چهار معیار اصلی استفاده شد که هر کدام به طور مختصر توضیح داده می‌شوند و فرمول ریاضی مربوط به هر کدام نیز بیان می‌شود. اولین معیار، R-Squared یا ضریب تعیین است که میزان توضیح‌پذیری واریانس داده‌های واقعی توسط مدل پیش‌بینی را نشان می‌دهد. فرمول محاسبه R^2 به صورت زیر است:

$$R^2 = 1 - \frac{SSE}{SST}$$

$$= 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}{\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \mu_y\right)^2}$$

$$= 1 - \frac{MSE}{Var(y)}$$

شکل ۱۹ فرمول محاسبه **R2Score**

در این فرمول، y_i مقادیر واقعی، ŷ_i مقادیر پیش‌بینی‌شده توسط مدل و Ȳ میانگین مقادیر واقعی هستند. مقدار R^2 در بازه‌ای از ۰ تا ۱ قرار دارد؛ هرچه مقدار R^2 به ۱ نزدیک‌تر باشد، نشان‌دهنده عملکرد بسیار خوب مدل است، در حالی که مقدار نزدیک به ۰ بیانگر عملکرد ضعیف مدل است.

دومین معیار، RMSE یا ریشه میانگین مربعات خطا، یکی از معیارهای رایج برای ارزیابی دقت مدل‌های پیش‌بینی به شمار می‌آید. فرمول RMSE به صورت زیر است:

$$\sqrt{\frac{{}^{2}(\hat{}_{i}y - {}_{i}y)\sum}{n}} = RMSE$$

شکل ۲۰ فرمول محاسبه **RMSE**

در اینجا، n تعداد نمونه‌ها است. RMSE میزان خطای پیش‌بینی مدل را به صورت عددی نشان می‌دهد و مقادیر کوچکتر این معیار نشان‌دهنده دقت بالاتر مدل هستند.

سومین معیار، MAE یا میانگین خطای مطلق، میانگین اختلاف بین مقادیر واقعی و پیش‌بینی‌شده را اندازه‌گیری می‌کند. فرمول MAE به شکل زیر است:

$$\frac{|\hat{y}_i - y_i| \sum}{n} = MAE$$

شکل ۲۱ فرمول محاسبه **MAE**

این معیار به ما کمک می‌کند تا به طور مستقیم میزان انحراف پیش‌بینی‌های مدل از مقادیر واقعی را ارزیابی کنیم و مقادیر کوچکتر MAE نشان‌دهنده عملکرد بهتر مدل است.

چهارمین معیار، MAPE یا میانگین خطای درصدی مطلق، درصد خطای مطلق را نسبت به مقادیر واقعی نشان می‌دهد. فرمول MAPE به صورت زیر است:

$$100 \times \left(\frac{\frac{|\hat{y}_i - y_i|}{y_i} \sum}{n} \right) = MAPE$$

شکل ۲۲فرمول محاسبه **MAPE**

این معیار به‌ویژه در مقایسه مدل‌ها در داده‌هایی با مقیاس‌های مختلف بسیار مفید است و مقادیر کمتر آن نشان‌دهنده دقت بیشتر مدل می‌باشد.

پس از اجرای مدل‌ها و انجام پیش‌بینی‌ها، مقادیر این معیارها برای سه مدل LSTM، Bi-LSTM و GRU محاسبه شدند. در ادامه به بررسی مقادیر این معیارها برای هر یک از این مدل‌ها پرداخته می‌شود.

```
LSTM Metrics (Scaled): (0.010500598878088999, 0.014798056948874063, 0.9808241346953824, 16.1273119586567)
Bi-LSTM Metrics (Scaled): (0.009796678027410834, 0.013453091493021355, 0.984151441275923, 16.01950690974525)
GRU Metrics (Scaled): (0.007450785467391785, 0.010686406151708641, 0.9899997942837199, 16.07001694184868)
LSTM Metrics (Unscaled): (1.5914708369886656, 2.242793522624053, 0.980824134847048, 2.0643870916344413)
Bi-LSTM Metrics (Unscaled): (1.4847848898206115, 2.038950794403296, 0.9841514377121109, 1.926832315806592)
GRU Metrics (Unscaled): (1.1292412855179095, 1.619631823671901, 0.9899997931396907, 1.4681135233254263)
```

شکل ۲۳ دقت مدل های یادگیری عمیق در حالات مختلف

برای ارزیابی عملکرد مدل‌های استفاده‌شده، معیارهای MAE، RMSE، R-Squared و MAPE برای هر سه مدل LSTM، Bi-LSTM و GRU محاسبه شدند. این معیارها برای دو حالت داده‌های نرمال‌شده (اسکیل‌شده) و داده‌های بازگردانده‌شده به مقیاس واقعی بررسی شده‌اند.

ابتدا به تحلیل نتایج برای داده‌های نرمال‌شده می‌پردازیم. مدل LSTM با مقدار MAE برابر با ۰٫۰۱۰۵ توانست خطای مطلق بسیار کمی را ارائه دهد. این مقدار نشان می‌دهد که فاصله میان مقادیر پیش‌بینی‌شده و مقادیر واقعی بسیار اندک بوده است. RMSE برای این مدل برابر با ۰٫۰۱۴۷ بود که نشان‌دهنده میزان خطای مربعات متوسط کوچک و انحراف پایین پیش‌بینی‌ها از مقادیر واقعی است. مقدار R-Squared برابر

با ۰,۹۸۰۸، دقت بالای مدل را در توضیح واریانس داده‌ها نشان می‌دهد. MAPE نیز که خطای نسبی درصدی را بیان می‌کند، برای این مدل حدود ۱۶,۱۲ درصد به‌دست آمد که نشان‌دهنده عملکرد قابل قبول مدل LSTM است.

در مدل Bi-LSTM، مقادیر MAE و RMSE به ترتیب برابر با ۰,۰۰۹۷ و ۰,۰۱۳۴ به‌دست آمدند که هر دو مقدار نشان‌دهنده بهبود عملکرد نسبت به LSTM هستند. این بهبود، بیانگر قدرت بالای Bi-LSTM در کاهش خطاهای پیش‌بینی است. مقدار R-Squared برابر با ۰,۹۸۴۵ نشان‌دهنده دقت بالاتر مدل Bi-LSTM نسبت به LSTM است. علاوه بر این، مقدار MAPE نیز برابر با ۱۶,۰۱ درصد گزارش شد که اندکی بهتر از LSTM عمل کرده است.

مدل GRU با مقادیر MAE برابر با ۰,۰۰۷۴ و RMSE برابر با ۰,۰۱۰۶، کمترین خطا را در بین سه مدل داشت. مقدار R-Squared این مدل برابر با ۰,۹۸۹۹ بود که نشان می‌دهد GRU بهترین عملکرد را در توضیح واریانس داده‌ها داشته است. مقدار MAPE این مدل نیز برابر با ۱۶,۰۷ درصد بود که اگرچه اختلاف کمی با Bi-LSTM دارد، اما همچنان نشان‌دهنده دقت بالای این مدل است.

حال به تحلیل نتایج مربوط به داده‌های واقعی (بازگردانده‌شده به مقیاس اصلی) می‌پردازیم. مدل LSTM با مقدار MAE برابر با ۱,۵۹ عملکرد مناسبی در داده‌های واقعی داشت. مقدار RMSE برابر با ۲,۲۴ نشان‌دهنده میزان انحراف پیش‌بینی‌ها از مقادیر واقعی بود که همچنان در یک بازه قابل قبول قرار دارد. مقدار R-Squared برای این مدل برابر با ۰,۹۸۸۲ بود که نشان‌دهنده توانایی خوب این مدل در توضیح واریانس داده‌ها است. مقدار MAPE این مدل برابر با ۲۰,۶۴ درصد بود که نسبت به داده‌های اسکیل‌شده اندکی افزایش یافت.

در مدل Bi-LSTM، مقادیر MAE و RMSE به ترتیب برابر با ۱,۴۸ و ۲,۰۳ به‌دست آمدند. این مقادیر نشان می‌دهند که Bi-LSTM نسبت به LSTM عملکرد بهتری در داده‌های واقعی دارد. مقدار R-Squared این مدل برابر با ۰,۹۸۴۵ گزارش شد که بیانگر دقت بالای مدل در توضیح واریانس داده‌ها است. مقدار MAPE این مدل نیز ۱۹,۲۶ درصد بود که نسبت به LSTM بهبود داشت.

مدل GRU با مقادیر MAE برابر با ۱,۱۲ و RMSE برابر با ۱,۶۱، کمترین خطا را در بین سه مدل نشان داد. مقدار R-Squared این مدل برابر با ۰,۹۸۹۹ بود که بیانگر دقت بالای مدل در توضیح واریانس داده‌ها است. مقدار MAPE این مدل نیز برابر با ۱۴,۶۸ درصد بود که کمترین مقدار در میان سه مدل بود و نشان‌دهنده عملکرد برتر GRU است.

در نهایت، با توجه به تحلیل نتایج، مدل GRU بهترین عملکرد را در هر دو مجموعه داده (نرمال‌شده و واقعی) ارائه داد. این مدل با مقادیر کمتر MAE و RMSE و همچنین R-Squared بالاتر، به‌عنوان دقیق‌ترین مدل شناخته شد. Bi-LSTM نیز عملکرد بسیار خوبی داشت و به‌عنوان مدلی قوی با پیچیدگی کمتر نسبت به GRU مورد تأیید قرار گرفت. مدل LSTM اگرچه عملکرد ضعیف‌تری نسبت به دو مدل دیگر داشت، اما همچنان توانست نتایج قابل قبولی ارائه دهد و در شرایطی با منابع محاسباتی محدودتر، گزینه مناسبی محسوب می‌شود.

۲-۴. ARIMA

ARIMAچیست ؟

مدل ARIMA که به معنی مدل خودرگرسیو یکپارچه با میانگین متحرک است، یکی از مدل‌های مهم و پرکاربرد در تحلیل سری‌های زمانی به شمار می‌رود. این مدل برای پیش‌بینی داده‌های سری زمانی که دارای روند مشخص اما بدون الگوهای فصلی هستند، به‌کار گرفته می‌شود. ساختار ساده، انعطاف‌پذیری در مدلسازی رفتارهای مختلف سری زمانی، و قابلیت پیش‌بینی مؤثر باعث شده است که مدل ARIMA به یکی از انتخاب‌های اصلی برای تحلیل سری‌های زمانی تبدیل شود.

مدل ARIMA از سه مؤلفه اصلی تشکیل شده است. مؤلفه اول، بخش خودرگرسیو (AR)است که تأثیر مقادیر گذشته را بر مقدار فعلی مدل‌سازی می‌کند. در این بخش، فرض بر این است که مقدار فعلی سری زمانی را می‌توان به صورت ترکیبی خطی از مقادیر گذشته محاسبه کرد. مؤلفه دوم، بخش یکپارچه‌سازی یا تفاضل‌گیری ((Iاست. این بخش زمانی استفاده می‌شود که سری زمانی دارای روند مشخص یا غیرایستا باشد. در چنین مواردی، با انجام تفاضل‌گیری، داده‌ها به حالت ایستا تبدیل می‌شوند تا برای مدلسازی مناسب‌تر شوند. مؤلفه سوم، بخش میانگین متحرک ((MAاست که خطاهای پیش‌بینی گذشته را به صورت ترکیبی خطی در نظر می‌گیرد و به کاهش تأثیر این خطاها در پیش‌بینی کمک می‌کند.

یکی از ویژگی‌های کلیدی مدل ARIMA استفاده از پارامترهای p و d و q است. پارامتر p نشان‌دهنده تعداد مقادیر گذشته‌ای است که در پیش‌بینی استفاده می‌شود و به بخش خودرگرسیو مرتبط است. پارامتر d تعداد دفعات تفاضل‌گیری را تعیین می‌کند که برای ایستا کردن داده‌ها ضروری است. پارامتر q نیز تعداد خطاهای پیش‌بینی گذشته‌ای را نشان می‌دهد که در بخش میانگین متحرک مدل‌سازی می‌شوند.

برای استفاده از مدل ARIMA، ابتدا باید داده‌ها بررسی شوند تا مشخص شود که آیا ایستا هستند یا خیر. این کار معمولاً با استفاده از آزمون‌هایی مانند آزمون دیکی-فولر انجام می‌شود. اگر داده‌ها ایستا نباشند، باید از تفاضل‌گیری برای ایستا کردن آن‌ها استفاده کرد. سپس پارامترهای p و d و q تعیین

می‌شوند. این کار با استفاده از روش‌هایی مانند کمینه‌سازی معیار اطلاعاتی آکایکه (AIC)یا معیار اطلاعاتی بیزین (BIC) انجام می‌شود. در نهایت، مدل با استفاده از داده‌های آموزش تنظیم و برای پیش‌بینی داده‌های آینده استفاده می‌شود.

یکی از ویژگی‌های مهم ARIMA این است که این مدل فقط برای داده‌های غیر فصلی مناسب است. به همین دلیل، اگر داده‌ها دارای الگوهای فصلی باشند، این مدل نمی‌تواند به خوبی رفتار سری زمانی را پیش‌بینی کند. این محدودیت باعث شده است که مدل SARIMA به‌عنوان نسخه گسترش‌یافته ARIMA معرفی شود. SARIMA توانایی مدلسازی رفتارهای فصلی را نیز داراست و از اجزای اضافی برای این منظور استفاده می‌کند.

در محاسبات مدل ARIMA، ابتدا داده‌ها به صورت ایستا تنظیم می‌شوند و سپس با استفاده از پارامترهای تعیین‌شده، وابستگی‌های زمانی مدل‌سازی می‌شوند. پیش‌بینی با ترکیب مقادیر گذشته و خطاهای پیش‌بینی قبلی انجام می‌شود و مدل تلاش می‌کند تا الگوهای موجود در داده‌ها را شناسایی و پیش‌بینی‌های دقیق‌تری ارائه دهد. خروجی این مدل معمولاً مقادیر پیش‌بینی‌شده سری زمانی برای بازه‌های زمانی آینده است.

در پروژه ما، از مدل ARIMA برای تحلیل و پیش‌بینی قیمت نفت خام استفاده کردیم. ابتدا داده‌ها را بررسی کردیم تا مشخص شود که ایستا هستند یا خیر. با توجه به غیرایستا بودن داده‌ها، فرآیند تفاضل‌گیری را اعمال کردیم تا روندهای موجود حذف شوند و داده‌ها ایستا شوند. سپس با استفاده از معیار AIC بهترین مقادیر برای پارامترهای p و d و q انتخاب شدند. در نهایت، مدل ARIMA برای پیش‌بینی قیمت نفت خام آموزش داده شد. اگرچه این مدل در پیش‌بینی‌های کوتاه‌مدت عملکرد خوبی داشت، اما به دلیل عدم توانایی در مدلسازی رفتارهای فصلی، در داده‌های بلندمدت یا فصلی محدودیت‌هایی نشان داد.

SARIMA چیست ؟

همان‌طور که پیش‌تر درباره مدل ARIMA توضیح دادیم، این مدل شامل سه مؤلفه اصلی است: خودرگرسیو (AR)، تفاضل‌گیری (I)و میانگین متحرک .(MA)این مؤلفه‌ها به ترتیب مسئول مدلسازی وابستگی به مقادیر گذشته، حذف روند و مدلسازی خطاهای پیش‌بینی هستند. مدل SARIMA نیز ساختاری مشابه دارد، اما با این تفاوت که علاوه بر این مؤلفه‌های غیر فصلی، بخش‌هایی برای تحلیل رفتارهای فصلی داده‌ها به آن اضافه شده است. این افزوده‌ها مدل را قادر می‌سازند تا داده‌هایی را که دارای الگوهای تکرارشونده فصلی هستند، به‌طور دقیق‌تر مدلسازی کند.

در SARIMA، مؤلفه‌های فصلی به مدل اضافه می‌شوند که شامل بخش‌های خودرگرسیو فصلی، تفاضل‌گیری فصلی و میانگین متحرک فصلی هستند. مؤلفه خودرگرسیو فصلی (P)تأثیر مقادیر گذشته در دوره‌های مشابه فصلی (مانند ماه‌های یک سال) را در پیش‌بینی مقدار فعلی تحلیل می‌کند. برای مثال، این بخش به مدل امکان می‌دهد تا تأثیر قیمت نفت در ماه ژانویه سال‌های گذشته را در پیش‌بینی قیمت ژانویه امسال لحاظ کند. تفاضل‌گیری فصلی (D)به مدل کمک می‌کند تا روندهای تکرارشونده در بازه‌های فصلی را حذف کند و سری زمانی فصلی را به حالت ایستا تبدیل نماید. این تفاضل‌گیری شبیه به تفاضل‌گیری غیر فصلی در ARIMA است، اما در بازه‌های زمانی مشخص مانند هر ۱۲ ماه اعمال می‌شود. میانگین متحرک فصلی (Q)نیز خطاهای پیش‌بینی در دوره‌های فصلی گذشته را برای بهبود پیش‌بینی مقدار فعلی در نظر می‌گیرد.

طول دوره فصلی با پارامتر s تعیین می‌شود که نشان می‌دهد الگوهای فصلی سری زمانی هر چند وقت یک‌بار تکرار می‌شوند. برای مثال، در داده‌های ماهانه با الگوهای سالانه، مقدار s برابر با ۱۲ در نظر گرفته می‌شود.

اگر بخواهیم یک مقداری نحوه محاسبات را در این مدل شفاف تر کنیم باید بگوئیم که مانند ARIMA، در SARIMA نیز ابتدا داده‌ها بررسی می‌شوند تا مشخص شود که آیا ایستا هستند یا خیر. اگر داده‌ها دارای روند یا تغییرات سیستماتیک باشند، تفاضل‌گیری غیر فصلی انجام می‌شود. برای داده‌هایی که رفتارهای فصلی دارند، تفاضل‌گیری فصلی نیز به‌طور جداگانه اعمال می‌شود. این کار باعث می‌شود که داده‌ها هم از نظر کلی و هم از نظر فصلی به حالت ایستا برسند.

مدل SARIMA پس از ایستا کردن داده‌ها، با استفاده از ترکیب پارامترهای غیر فصلی و فصلی، سری زمانی را مدلسازی می‌کند. همان‌طور که برای ARIMA توضیح داده شد، بخش خودرگرسیو به مدلسازی مقادیر گذشته سری زمانی می‌پردازد، بخش تفاضل‌گیری روندها را حذف می‌کند و بخش میانگین متحرک خطاهای پیش‌بینی را تصحیح می‌کند. این فرآیند در SARIMA نیز مشابه است، اما برای رفتارهای فصلی نیز اعمال می‌شود. به این ترتیب، مدل نه‌تنها داده‌های کوتاه‌مدت و غیر فصلی را تحلیل می‌کند، بلکه به الگوهای تکرارشونده فصلی نیز توجه دارد.

در SARIMA، پارامترهای P، D و Q برای مدلسازی رفتارهای فصلی مشابه نقش پارامترهای p، d و q در رفتارهای غیر فصلی هستند. با این تفاوت که این پارامترها تنها به مقادیر و خطاهای مربوط به دوره‌های فصلی توجه دارند. به عنوان مثال، پارامتر P تعداد مقادیر گذشته در بازه‌های فصلی مشابه (مانند ماه‌های مشابه در سال‌های مختلف) را که برای پیش‌بینی استفاده می‌شوند، تعیین می‌کند. D تعداد دفعات

تفاضل‌گیری فصلی را مشخص می‌کند و Q تعداد خطاهای فصلی مورد استفاده در میانگین متحرک فصلی را تعیین می‌کند.

در نهایت وقتی میخواهیم مدل را تنظیم و از آن استفاده کنیم، ابتدا داده‌های سری زمانی آماده می‌شوند. داده‌ها باید ایستا شوند، که این کار از طریق تفاضل‌گیری غیر فصلی و فصلی انجام می‌شود. پس از ایستا کردن داده‌ها، پارامترهای مدل تعیین می‌شوند. این پارامترها شامل مقادیر p، d، q برای رفتارهای غیر فصلی و مقادیر P، D، Q، s برای رفتارهای فصلی هستند. انتخاب این پارامترها به‌گونه‌ای است که مدل بتواند بهترین تطابق را با داده‌ها داشته باشد. برای یافتن بهترین ترکیب پارامترها، معمولاً از معیارهای AIC یا BIC استفاده می‌شود که تعادلی بین دقت مدل و پیچیدگی آن برقرار می‌کنند.

مدل SARIMA پس از تنظیم پارامترها، آماده پیش‌بینی است. پیش‌بینی‌ها می‌توانند به صورت تک مرحله‌ای یا چند مرحله‌ای انجام شوند. در پیش‌بینی تک مرحله‌ای، مدل تنها مقدار بعدی سری زمانی را پیش‌بینی می‌کند. در پیش‌بینی چند مرحله‌ای، مدل به صورت بازگشتی عمل می‌کند، یعنی ابتدا مقدار اول پیش‌بینی می‌شود، سپس این مقدار به تاریخچه داده‌ها اضافه می‌شود و مدل مقدار بعدی را پیش‌بینی می‌کند. این فرآیند برای تمامی مراحل پیش‌بینی تکرار می‌شود.

در پروژه ما، مدل SARIMA برای پیش‌بینی قیمت نفت خام استفاده شد. ابتدا داده‌ها بررسی شدند و مشخص شد که دارای الگوهای فصلی هستند. تفاضل‌گیری غیر فصلی و فصلی برای ایستا کردن داده‌ها انجام شد. سپس با استفاده از روش‌های بهینه‌سازی، مقادیر بهینه برای تمامی پارامترهای فصلی و غیر فصلی تعیین شدند. مدل نهایی آموزش داده شد و برای پیش‌بینی قیمت نفت خام در آینده مورد استفاده قرار گرفت. نتایج نشان داد که SARIMA به‌خوبی توانست رفتارهای فصلی و غیر فصلی داده‌ها را مدلسازی کند و پیش‌بینی‌های دقیقی ارائه دهد.

تفاوت‌های اصلی این دو مدل:

مدل ARIMA برای داده‌هایی طراحی شده است که رفتارهای غیر فصلی دارند. این مدل قادر است روندها، نویزهای تصادفی و تغییرات بلندمدت داده‌ها را تحلیل و پیش‌بینی کند. ساختار ARIMA به سه مؤلفه اصلی تقسیم می‌شود: مؤلفه خودرگرسیو که تأثیر مقادیر گذشته را بر مقدار فعلی بررسی می‌کند، مؤلفه تفاضل‌گیری که با حذف روند، داده‌ها را ایستا می‌کند، و مؤلفه میانگین متحرک که خطاهای پیش‌بینی قبلی را برای بهبود پیش‌بینی‌های فعلی در نظر می‌گیرد. این سه مؤلفه به ARIMA کمک می‌کنند تا داده‌های سری زمانی را به‌خوبی مدلسازی کند، اما این مدل برای داده‌هایی که رفتارهای تکرارشونده فصلی دارند مناسب نیست.

در مقابل، مدل SARIMA نسخه گسترش‌یافته ARIMA است که علاوه بر رفتارهای غیر فصلی، توانایی مدلسازی رفتارهای فصلی را نیز دارد. در بسیاری از سری‌های زمانی واقعی، مانند فروش فصلی یا تغییرات دما، الگوهایی وجود دارند که در بازه‌های زمانی مشخص، مانند هر ماه یا هر سال، تکرار می‌شوند. SARIMA با افزودن مؤلفه‌های فصلی به مدل ARIMA، این رفتارهای دوره‌ای را نیز در نظر می‌گیرد. مؤلفه‌های فصلی SARIMA شامل خودرگرسیو فصلی، تفاضل‌گیری فصلی و میانگین متحرک فصلی هستند که هرکدام به ترتیب تأثیر مقادیر گذشته در دوره‌های مشابه، حذف روندهای تکرارشونده، و اصلاح خطاهای پیش‌بینی در بازه‌های فصلی را تحلیل می‌کنند. همچنین، در SARIMA پارامتر مهمی به نام طول دوره فصلی وجود دارد که مشخص می‌کند الگوهای تکرارشونده در چه بازه زمانی رخ می‌دهند، مثلاً هر ۱۲ ماه برای داده‌های ماهانه.

تفاوت اصلی این دو مدل در توانایی SARIMA برای تحلیل و پیش‌بینی داده‌های فصلی نهفته است. ARIMA برای داده‌هایی که الگوهای فصلی ندارند بسیار مناسب است، زیرا تمام تمرکز آن بر روی روندهای کلی و نویزهای تصادفی است. اما SARIMA به دلیل داشتن مؤلفه‌های فصلی می‌تواند داده‌هایی را که شامل تغییرات دوره‌ای و تکرارشونده هستند نیز به‌خوبی مدلسازی کند. این ویژگی باعث می‌شود که SARIMA در پیش‌بینی داده‌های واقعی مانند تغییرات آب‌وهوایی، فروش فصلی یا قیمت نفت خام که معمولاً الگوهای فصلی دارند، بسیار مؤثرتر عمل کند.

مزایا و محدودیت های مدل ARIMAرا ذکر کنید.

یکی از بزرگ‌ترین مزایای مدل ARIMA، سادگی و کارایی آن است. این مدل به دلیل ساختار ساده خود، برای تحلیل سری‌های زمانی بدون رفتارهای فصلی بسیار مناسب است و به راحتی می‌توان آن را در بسیاری از موارد به کار گرفت. ARIMA از سه مؤلفه اصلی تشکیل شده است: مؤلفه خودرگرسیو (AR)، مؤلفه تفاضل‌گیری (I)، و مؤلفه میانگین متحرک .(MA)مؤلفه خودرگرسیو وابستگی مقادیر فعلی سری زمانی به مقادیر گذشته را مدلسازی می‌کند. مؤلفه تفاضل‌گیری برای حذف روندهای موجود در داده‌ها به کار می‌رود و باعث می‌شود که سری زمانی به حالت ایستا تبدیل شود. در نهایت، مؤلفه میانگین متحرک با در نظر گرفتن خطاهای پیش‌بینی‌های قبلی، دقت پیش‌بینی‌های مدل را افزایش می‌دهد. این سه مؤلفه به مدل ARIMA امکان می‌دهند که روندها و نویزهای موجود در داده‌ها را به خوبی تحلیل کند و پیش‌بینی‌هایی دقیق ارائه دهد.

یکی دیگر از مزایای این مدل، توانایی آن در تحلیل داده‌های غیر ایستا است. در بسیاری از سری‌های زمانی، داده‌ها دارای روندهایی هستند که مانع از ایستایی آن‌ها می‌شود. مدل ARIMA با استفاده از مؤلفه تفاضل‌گیری، این روندها را حذف کرده و داده‌ها را به حالت ایستا تبدیل می‌کند تا بتوان تحلیل و پیش‌بینی

دقیقی انجام داد. همچنین، این مدل نیازی به ورودی‌های اضافی ندارد و تنها بر اساس داده‌های سری زمانی عمل می‌کند. این ویژگی باعث می‌شود که ARIMA در بسیاری از موارد، ابزار ساده و موثری برای پیش‌بینی باشد.

با وجود این مزایا، مدل ARIMA محدودیت‌هایی نیز دارد که باید به آن‌ها توجه شود. یکی از مهم‌ترین محدودیت‌های این مدل، ناتوانی در مدلسازی رفتارهای فصلی است. بسیاری از سری‌های زمانی واقعی، مانند تغییرات فروش یا دما، دارای الگوهای فصلی هستند که در بازه‌های زمانی مشخص تکرار می‌شوند. ARIMA به دلیل نداشتن مؤلفه‌های فصلی، نمی‌تواند این الگوها را شناسایی و پیش‌بینی کند و در چنین مواردی دقت پیش‌بینی آن کاهش می‌یابد. در چنین شرایطی، استفاده از مدل SARIMA که نسخه گسترش‌یافته ARIMA است، مناسب‌تر خواهد بود.

یکی دیگر از محدودیت‌های ARIMA، وابستگی آن به ایستایی داده‌ها است. اگر داده‌ها به درستی ایستا نشوند، مدل نمی‌تواند پیش‌بینی‌های دقیقی ارائه دهد. علاوه بر این، تنظیم پارامترهای مدل، یعنی تعداد مقادیر گذشته (p)، تعداد دفعات تفاضل‌گیری (d)، و تعداد خطاهای پیش‌بینی قبلی (q)، به دقت و آزمون‌های متعددی نیاز دارد. انتخاب نادرست این پارامترها می‌تواند عملکرد مدل را به شدت تحت تأثیر قرار دهد.

از دیگر محدودیت‌های ARIMA می‌توان به کاهش دقت در پیش‌بینی‌های بلندمدت اشاره کرد. این مدل برای پیش‌بینی‌های کوتاه‌مدت بسیار دقیق عمل می‌کند، اما در پیش‌بینی‌های بلندمدت، تجمع خطاها باعث کاهش دقت پیش‌بینی‌ها می‌شود. همچنین، این مدل تنها بر داده‌های تک‌متغیره تمرکز دارد و نمی‌تواند روابط بین متغیرهای مختلف را در تحلیل خود در نظر بگیرد. به همین دلیل، برای مسائل پیچیده‌تر که متغیرهای متعددی دخیل هستند، ممکن است ARIMA گزینه مناسبی نباشد.

در نهایت، ARIMA نسبت به داده‌های پرت نیز حساس است. وجود داده‌های پرت می‌تواند دقت مدل را کاهش داده و نتایج پیش‌بینی را تحت تأثیر قرار دهد. علاوه بر این، اگر داده‌های سری زمانی بسیار بزرگ باشند، محاسبات مربوط به این مدل زمان‌بر خواهد بود و کارایی آن کاهش می‌یابد.

مدل ARIMAپارامتر هایی دارد، مفهوم ریاضی این مدل را با ذکر پارامتر ها شرح دهید.

مؤلفه خودرگرسیو (AR) و پارامتر p : این مؤلفه نشان‌دهنده تأثیر مقادیر گذشته سری زمانی بر مقدار فعلی است. تعداد مقادیر گذشته‌ای که در پیش‌بینی دخیل هستند، با پارامتر pppمشخص می‌شود. به‌طور ریاضی، این بخش به صورت زیر بیان می‌شود:

$$_t\epsilon + {}_{t-p}\phi_p y + \cdots + {}_{t-2}\phi_2 y + {}_{t-1}\phi_1 y = {}_t y$$

در این معادله $_ty$ به معنی مقدار فعلی سری زمانی است. y_p به معنی مقادیر گذشته سری زمانی هستند. ضرایب موجود نیز ضرایب خودرگرسیو هستند که تأثیر هر مقدار گذشته را تعیین می‌کنند. ϵt نویز یا خطای تصادفی است.

پارامتر p تعیین می‌کند که چند مقدار گذشته در مدل لحاظ شود. اگر مقدار p خیلی بزرگ انتخاب شود، مدل بیش از حد پیچیده می‌شود و ممکن است دچار بیش‌برازش شود.

۲. مؤلفه تفاضل‌گیری (I)و پارامتر d: این مؤلفه برای ایستا کردن سری زمانی به‌کار می‌رود. اگر سری زمانی دارای روند باشد، تفاضل‌گیری این روند را حذف کرده و داده‌ها را به حالت ایستا تبدیل می‌کند. پارامتر d d تعداد دفعات تفاضل‌گیری را مشخص می‌کند. تفاضل‌گیری به صورت زیر تعریف می‌شود:

$$_{t-1}y - _ty = {}'_ty$$

برای سری‌هایی که نیاز به بیش از یک تفاضل‌گیری دارند، فرآیند به صورت تکراری انجام می‌شود.

پارامتر d تعیین می‌کند که چند بار تفاضل‌گیری لازم است تا سری زمانی ایستا شود. انتخاب نادرست این پارامتر ممکن است منجر به حذف اطلاعات مهم یا باقی ماندن روند در داده‌ها شود.

۳. مؤلفه میانگین متحرک (MA) و پارامترq : این مؤلفه نشان‌دهنده تأثیر خطاهای پیش‌بینی گذشته بر مقدار فعلی سری زمانی است. تعداد خطاهای پیش‌بینی گذشته که در مدل لحاظ می‌شوند، با پارامتر qمشخص می‌شود. به‌طور ریاضی، این بخش به صورت زیر بیان می‌شود:

$$_t\epsilon + _{t-q}\theta_q\epsilon + \cdots + _{t-2}\theta_2\epsilon + _{t-1}\theta_1\epsilon = _ty$$

ε ها در اینجا به معنی خطاهای پیش‌بینی قبلی هستند. θ ها به معنی ضرایب میانگین متحرک هستند که تأثیر هر خطای گذشته را تعیین می‌کنند.

پارامتر q تعیین می‌کند که چند خطای پیش‌بینی گذشته در مدل لحاظ شوند. این پارامتر به مدل کمک می‌کند تا نوسانات سری زمانی را به‌خوبی مدلسازی کند.

مدل ARIMA ترکیبی از این سه مؤلفه است و به صورت کلی به شکل زیر بیان می‌شود:

$$_t\epsilon + _{t-q}\theta_q\epsilon + \cdots + _{t-2}\theta_2\epsilon + _{t-1}\theta_1\epsilon + _{t-p}\phi_p y + \cdots + _{t-2}\phi_2 y + _{t-1}\phi_1 y = _ty$$

این مدل بر روی داده‌های تفاضل‌گیری شده اعمال می‌شود. بنابراین، اگر تفاضل‌گیری با مقدار dddاعمال شده باشد، سری زمانی ایستا شده ابتدا محاسبه می‌شود و سپس مؤلفه‌های خودرگرسیو و میانگین متحرک بر روی آن اعمال می‌شوند.

پارامتر های بهینه ی این مدل را بدست آورده و گزارش کنید

```
from statsmodels.tsa.stattools import adfuller
result = adfuller(train_data['Adj Close'])
print(f"ADF Statistic: {result[0]}")
print(f"p-value: {result[1]}")
print(f"Critical Values: {result[4]}")

if result[1] < 0.05:
    print("The series is stationary.")
else:
    print("The series is not stationary.")
```

```
ADF Statistic: -1.3175200901257018
p-value: 0.6211192230668825
Critical Values: {'1%': -3.4320679607261453, '5%': -2.86229910063996, '10%': -2.5671740791451567}
The series is not stationary.
```

شکل ۲۴انجام تست **ADF**

در این بخش از پروژه، هدف بررسی ایستایی داده‌های سری زمانی مربوط به قیمت نفت خام بود. ایستایی در سری‌های زمانی به این معناست که ویژگی‌های آماری داده‌ها، مانند میانگین، واریانس و همبستگی، در طول زمان ثابت باقی بمانند. این ویژگی به‌ویژه برای بسیاری از مدل‌های سری زمانی، از جمله مدل ARIMA، بسیار اهمیت دارد. این مدل‌ها فرض می‌کنند که داده‌ها ایستا هستند تا بتوانند روندهای زمانی را بهتر مدلسازی کرده و پیش‌بینی کنند. اگر داده‌ها ایستا نباشند، ممکن است نتایج مدل دچار خطا شود یا عملکرد مطلوبی نداشته باشد.

برای بررسی ایستایی داده‌ها، از آزمون دیکی-فولر افزوده (ADF)استفاده شد. این آزمون یکی از روش‌های استاندارد برای ارزیابی ایستایی سری‌های زمانی است و به ما کمک می‌کند تا مشخص کنیم آیا سری زمانی دارای روند است یا خیر. در این آزمون، فرض صفر به معنای غیراستایی داده‌ها و فرض مقابل به معنای ایستایی داده‌ها است. به عبارت ساده‌تر، اگر داده‌ها دارای روند زمانی باشند یا واریانس آن‌ها در طول زمان تغییر کند، داده‌ها غیراستا در نظر گرفته می‌شوند.

برای اجرای آزمون، سری زمانی قیمت نفت خام (Adj Close)که از مجموعه داده‌های ما استخراج شده بود، مورد بررسی قرار گرفت. نتیجه آزمون دیکی-فولر افزوده به ما سه مقدار اصلی ارائه داد: آماره آزمون، مقدار احتمالی یا p-value، و مقادیر بحرانی در سطوح مختلف اطمینان.

در این آزمایش، مقدار آماره آزمون برابر با -۳۱۷۵.۱ بود. این مقدار در مقایسه با مقادیر بحرانی ارائه‌شده در سطوح ۱٪، ۵٪ و ۱۰٪ (به ترتیب -۳,۴۳۰۲، -۲,۸۶۲۳ و -۲,۵۶۷۱) نشان داد که مقدار آماره آزمون از این مقادیر بحرانی بزرگ‌تر است. علاوه بر این، مقدار احتمالی یا p-value آزمون برابر با ۰,۶۲۱۱ بود، که این مقدار از آستانه ۰,۰۵ بیشتر است. به عبارت دیگر، این نتایج به ما نشان داد که فرض صفر آزمون (غیراستایی داده‌ها) رد نمی‌شود.

بنابراین، نتایج این آزمون به‌وضوح نشان داد که داده‌های سری زمانی ما ایستا نیستند. این به این معناست که این داده‌ها دارای روند هستند یا واریانس آن‌ها در طول زمان تغییر می‌کند. چنین داده‌هایی برای مدل‌سازی با مدل ARIMA یا مدل‌های مشابه مناسب نیستند، زیرا این مدل‌ها نیازمند داده‌های ایستا هستند.

```
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=13112.198, Time=0.67 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=13114.977, Time=0.06 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=13103.285, Time=0.13 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=13102.068, Time=0.16 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=13113.223, Time=0.06 sec
 ARIMA(0,1,2)(0,0,0)[0] intercept   : AIC=13099.774, Time=0.36 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=13070.265, Time=0.67 sec
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=13043.542, Time=1.72 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=13060.941, Time=0.80 sec
 ARIMA(3,1,2)(0,0,0)[0] intercept   : AIC=13038.415, Time=1.09 sec
 ARIMA(3,1,1)(0,0,0)[0] intercept   : AIC=13037.178, Time=0.92 sec
 ARIMA(3,1,0)(0,0,0)[0] intercept   : AIC=13038.639, Time=0.33 sec
 ARIMA(4,1,1)(0,0,0)[0] intercept   : AIC=13038.854, Time=1.91 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=13095.414, Time=0.28 sec
 ARIMA(4,1,0)(0,0,0)[0] intercept   : AIC=13038.316, Time=0.59 sec
 ARIMA(4,1,2)(0,0,0)[0] intercept   : AIC=13039.342, Time=3.22 sec
 ARIMA(3,1,1)(0,0,0)[0]             : AIC=13035.517, Time=0.43 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=13059.291, Time=0.47 sec
 ARIMA(3,1,0)(0,0,0)[0]             : AIC=13036.949, Time=0.24 sec
 ARIMA(4,1,1)(0,0,0)[0]             : AIC=13037.196, Time=2.05 sec
 ARIMA(3,1,2)(0,0,0)[0]             : AIC=13036.760, Time=0.43 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=13093.655, Time=0.10 sec
 ARIMA(2,1,2)(0,0,0)[0]             : AIC=13041.880, Time=0.92 sec
 ARIMA(4,1,0)(0,0,0)[0]             : AIC=13036.641, Time=0.39 sec
 ARIMA(4,1,2)(0,0,0)[0]             : AIC=13037.687, Time=1.97 sec

Best model:  ARIMA(3,1,1)(0,0,0)[0]
Total fit time: 19.979 seconds
Optimal ARIMA Order: (3, 1, 1)
```

شکل ۲۵پارامترهای انتخاب شده برای **ARIMA**

در این بخش از پروژه، ما به دنبال یافتن بهترین پارامترهای مدل ARIMA برای پیش‌بینی سری زمانی بودیم. برای انجام این کار، از روش خودکار auto_arima که بخشی از کتابخانه pmdarima است، استفاده

کردیم. این ابزار به طور خودکار ترکیبات مختلفی از پارامترهای مدل ARIMA را آزمایش می‌کند و با استفاده از معیار اطلاعات آکاییک (AIC)بهترین ترکیب پارامترها را انتخاب می‌کند. معیار AIC به عنوان یک ابزار آماری برای مقایسه مدل‌ها به کار می‌رود و مدلی که کمترین مقدار AIC را دارد، به عنوان بهترین مدل انتخاب می‌شود. این معیار با در نظر گرفتن دقت مدل و همچنین پیچیدگی آن، از ایجاد مدل‌های بیش از حد پیچیده یا اورفیت جلوگیری می‌کند.

برای شروع، داده‌های مربوط به قیمت بسته‌شده اصلاح‌شده (Adj Close)به عنوان ورودی اصلی مدل انتخاب شد. علاوه بر این، از متغیرهای کمکی مانند قیمت باز شدن (Open)، بیشینه (High)، کمینه (Low)و حجم معاملات (Volume)به عنوان داده‌های کمکی یا متغیرهای خارجی (Exogenous)برای بهبود دقت پیش‌بینی استفاده شد. این متغیرها به مدل کمک می‌کنند تا تأثیرات خارجی و پیچیدگی‌های بیشتری از داده‌ها را در نظر بگیرد.

برای یافتن مقادیر بهینه پارامترها، بازه‌های مناسب برای هر کدام تعریف شد. به عنوان مثال، مقدار p و q در بازه ۱ تا ۵ و مقدار d با حداکثر مقدار ۴ تنظیم شد. این بازه‌ها به گونه‌ای انتخاب شدند که مدل بتواند تمامی ترکیبات ممکن را بررسی کند، اما در عین حال سرعت اجرای مدل کاهش نیابد.

برای بررسی ایستایی داده‌ها، از آزمون KPSS استفاده شد. اگر سری زمانی ایستا نباشد، مدل تعداد دفعات لازم تفاضل‌گیری را به صورت خودکار تعیین می‌کند. در اینجا، تفاضل‌گیری به مدل کمک می‌کند تا روندهای احتمالی در داده‌ها حذف شده و سری زمانی به حالت ایستا تبدیل شود.

همچنین، تنظیماتی مانند stepwise=True برای بهینه‌سازی فرآیند جستجو و افزایش سرعت آن به صورت مرحله‌ای استفاده شد. در این تنظیم، مدل به جای جستجوی جامع تمامی ترکیبات، به صورت مرحله‌ای و با اولویت‌دهی به مقادیر محتمل‌تر، جستجو را انجام می‌دهد. علاوه بر این، گزینه suppress_warnings=True برای خاموش کردن هشدارهای غیرضروری و جلوگیری از ایجاد مزاحمت در فرآیند اجرا اعمال شد.

در نتیجه اجرای کد، تمامی ترکیبات ممکن از پارامترهای ARIMA بررسی شد و بهترین مدل با توجه به مقدار کمینه AIC انتخاب شد. همان‌طور که در خروجی مشاهده می‌شود، مدل ARIMA با پارامترهای (۳, ۱, ۱) به عنوان مدل بهینه انتخاب شد. این پارامترها به این معنا هستند که:

مقدار داده‌های فعلی به سه مقدار قبلی خود وابسته است

داده‌ها یک بار تفاضل‌گیری شده‌اند تا ایستا شوند

خطاهای پیش‌بینی مدل از یک گام قبلی استفاده می‌کنند

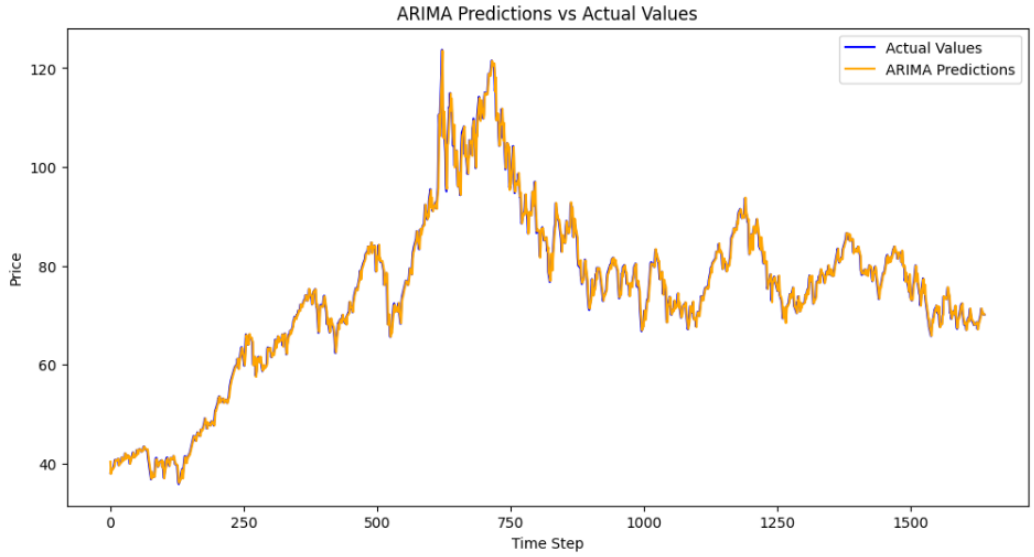
این فرآیند به ما اجازه داد تا مدلی ایجاد کنیم که نه تنها دقت پیش‌بینی بالایی داشته باشد، بلکه ساده و قابل اعتماد باشد. به طور کلی، انتخاب پارامترها بر اساس معیار AIC باعث می‌شود تا مدل در عین کاهش پیچیدگی، توانایی کافی برای مدلسازی و پیش‌بینی دقیق داده‌های سری زمانی را داشته باشد.

زمان کل اجرای فرآیند، حدود ۲۰ ثانیه گزارش شد که نشان‌دهنده بهینه‌سازی مناسب در جستجوی پارامترها بود. انتخاب مدل بهینه ARIMA با این پارامترها نشان داد که داده‌ها رفتارهای مشخصی دارند که به درستی توسط این ترکیب مدلسازی می‌شوند.

ضمن ارائه ی جدولی مشابه جدول شماره ،۶نتایج را با نتایج داخل مقاله مقایسه کنید

```
Processing ARIMA predictions: 100%|██████████| 1640/1640 [19:52<00:00,  1.38it/s]
ARIMA Metrics (Unscaled): (0.8697098477810211, 1.3583054514301331, 0.9936417162303871, 1.1364695957188116)
```

شکل ۲۶ دقت مدل **ARIMA**



شکل ۲۷ مقایسه خروجی های مدل **ARIMA** با مقادیر واقعی

در این بخش از پروژه، مدل ARIMA برای پیش‌بینی قیمت نفت خام مورد استفاده قرار گرفت و نتایج به‌دست‌آمده تحلیل شد. مدل ARIMA یکی از مدل‌های کلاسیک در تحلیل سری‌های زمانی است که بر پایه سه مؤلفه اصلی طراحی شده است: بخش خودبازگشتی (AR)، بخش تفاضلی (I)و بخش میانگین متحرک (MA). این مدل به‌طور گسترده‌ای برای داده‌هایی استفاده می‌شود که رفتارهای فصلی ندارند و به دنبال شناسایی روندها و پیش‌بینی الگوهای کوتاه‌مدت هستند.

در مرحله آموزش، از داده‌های نرمال‌شده برای آموزش مدل استفاده شد و با به‌کارگیری الگوریتم خودکار انتخاب پارامترهای ARIMA، بهترین مقادیر برای پارامترهای p و d و q شناسایی شدند. این مقادیر به ترتیب به معنای تعداد تأخیرهای خودبازگشتی، درجه تفاضل‌گیری و تعداد تأخیرهای خطای میانگین متحرک هستند. بر اساس خروجی الگوریتم، بهترین مقادیر برای این پارامترها برابر با (p=3)، (d=1) و (q=1)بودند. این مقادیر نشان‌دهنده ساختاری هستند که در آن سری زمانی با استفاده از سه مقدار گذشته خود، یک مرتبه تفاضل‌گیری و خطاهای پیش‌بینی‌های قبلی مدل‌سازی می‌شود.

در این مرحله، مدل با داده‌های آموزشی تطبیق داده شد و سپس برای پیش‌بینی داده‌های آزمایشی مورد استفاده قرار گرفت. در نهایت، نتایج پیش‌بینی‌شده مدل با مقادیر واقعی مقایسه شد. همان‌طور که در نمودار ارائه‌شده مشاهده می‌شود، مدل ARIMA توانسته است روند کلی قیمت نفت خام را با دقت مناسبی پیش‌بینی کند. پیش‌بینی‌ها و مقادیر واقعی در بسیاری از نقاط همپوشانی قابل‌توجهی دارند، که نشان‌دهنده توانایی مدل در شناسایی الگوهای اصلی در داده‌ها است.

برای ارزیابی عملکرد مدل، معیارهای استانداردی شامل میانگین خطای مطلق (MAE)، ریشه میانگین مربعات خطا (RMSE)، ضریب تعیین (R-squared)و میانگین درصد خطای مطلق (MAPE)محاسبه شدند. مقادیر به‌دست‌آمده به شرح زیر هستند:

- میانگین خطای مطلق: ۰,۸۶۹۷
- ریشه میانگین مربعات خطا: ۱,۳۵۸۳
- ضریب تعیین: ۰,۹۹۳۶
- میانگین درصد خطای مطلق: ۱,۱۳۶۴٪

این مقادیر نشان‌دهنده عملکرد مطلوب مدل ARIMA در پیش‌بینی سری‌های زمانی هستند. به‌ویژه، مقدار ضریب تعیین نزدیک به ۱ حاکی از این است که مدل توانسته است بخش عمده‌ای از واریانس داده‌ها را توضیح دهد. همچنین، مقدار پایین MAPE نشان‌دهنده این است که خطای نسبی پیش‌بینی‌های مدل نسبت به مقادیر واقعی بسیار کم بوده است.

در مقایسه با مدل‌های دیپ لرنینگ مانند LSTM، Bi-LSTM و GRU که در بخش قبلی مورد بررسی قرار گرفتند، مدل ARIMA در شناسایی روندهای کلی و الگوهای ساده داده‌ها عملکرد قابل قبولی داشت. با این حال، مدل‌های دیپ لرنینگ به دلیل توانایی‌شان در یادگیری روابط غیرخطی و پیچیده در داده‌ها، در شناسایی تغییرات ناگهانی و نویزهای غیرمنتظره عملکرد بهتری ارائه دادند. به عنوان مثال، مدل LSTM

توانست تغییرات جزئی‌تر و رفتارهای غیرخطی داده‌ها را بهتر شناسایی کند و خطای کمتری در پیش‌بینی‌ها داشته باشد.

ابتدا، مدل ARIMA که یک مدل کلاسیک خطی است، در پیش‌بینی داده‌ها عملکرد نسبتاً ضعیف‌تری نسبت به مدل‌های یادگیری عمیق نشان داد. این مدل به دلیل ماهیت خطی خود، قادر به مدلسازی روابط غیرخطی و وابستگی‌های بلندمدت در داده‌ها نیست. در پروژه ما، معیارهای ارزیابی عملکرد ARIMA شامل MAE، RMSE و MAPE به مراتب بالاتر از مدل‌های یادگیری عمیق بودند. به‌عنوان مثال، مقدار MAPE مدل ARIMA نسبت به سایر مدل‌ها بیشتر بود که این مسئله نشان‌دهنده دقت پایین‌تر آن در پیش‌بینی مقادیر واقعی است. این نتایج با مقاله مطابقت دارد، جایی که ARIMA تنها در داده‌های ساده و ایستا عملکرد مطلوبی داشت اما در داده‌های پیچیده‌تر و غیرخطی عملکرد مناسبی نداشت.

در مقابل، مدل‌های یادگیری عمیق مانند GRU، LSTM و Bi-LSTM عملکرد بهتری را در پیش‌بینی سری زمانی نشان دادند. این مدل‌ها به دلیل توانایی در یادگیری روابط غیرخطی و وابستگی‌های طولانی‌مدت، نتایج دقیق‌تری ارائه کردند. مدل GRU به‌طور خاص، در بین مدل‌های بررسی‌شده، کمترین مقدار MAPE را داشت که نشان‌دهنده توانایی بالای آن در پیش‌بینی مقادیر واقعی است. ساختار ساده‌تر GRU نسبت به LSTM باعث شده است که این مدل کارآمدتر باشد و همچنان دقت بالایی را ارائه دهد. مدل LSTM نیز، به دلیل ساختار پیچیده‌تر خود، توانایی قابل‌توجهی در یادگیری وابستگی‌های زمانی داشت و نتایج خوبی به‌دست آورد. Bi-LSTM، اگرچه کمی ضعیف‌تر از GRU و LSTM عمل کرد، اما همچنان عملکرد بهتری نسبت به ARIMA داشت. این مدل به دلیل توانایی در پردازش داده‌ها به صورت دوجهته، توانست اطلاعات بیشتری را از داده‌ها استخراج کند.

هنگامی که این نتایج با مقاله مقایسه شدند، روند مشابهی مشاهده شد. در مقاله نیز، مدل‌های GRU و LSTM به‌طور مداوم عملکرد بهتری نسبت به ARIMA داشتند. به‌عنوان مثال، در مقاله، مقدار MAPE مدل GRU به حدود ۲,۴۲ درصد رسید، در حالی که مقدار MAPE مدل ARIMA در برخی موارد بالای ۳۰ درصد گزارش شده بود. این تفاوت‌ها نشان‌دهنده مزیت استفاده از مدل‌های یادگیری عمیق در تحلیل داده‌های پیچیده و غیرخطی است.

علاوه بر این، در مقاله تأکید شده است که مدل ARIMA برای داده‌های ایستا و ساده مناسب است، اما برای داده‌هایی که شامل روندها، الگوهای فصلی و روابط غیرخطی هستند، مدل‌های یادگیری عمیق مانند GRU و LSTM گزینه‌های بسیار بهتری محسوب می‌شوند. این نتایج با پروژه ما مطابقت دارد، جایی که استفاده از GRU و LSTM باعث کاهش چشمگیر خطا و بهبود دقت پیش‌بینی‌ها شد.

جدول**6** مقایسه نتایج بدست آمده از مدل های مختلف

Model	MAE (مقاله)	RMSE (مقاله)	R-Squared (مقاله)	MAPE (مقاله)	MAE (ما)	RMSE (ما)	R-Squared (ما)	MAPE (ما)
LSTM	667.64	896.79	0.9425	3.9412	1.59	2.24	0.9882	2.06
Bi-LSTM	878.41	1108.92	0.9120	5.2493	1.48	2.03	0.9845	1.92
GRU	531.05	718.00	0.9631	3.0547	1.12	1.61	0.9899	1.46
ARIMA	4700.36	5817.17	-0.011	35.95	0.87	1.35	0.9936	1.13

با توجه به اینکه مقیاس داده‌های مورد استفاده در پروژه ما و مقاله به طور قابل توجهی متفاوت است، مقایسه نتایج نیاز به دقت بیشتری دارد. در مقاله، مقادیر داده‌های سری زمانی در حدود ۱۰ برابر بزرگ‌تر از مقادیر مورد استفاده در پروژه ما هستند. این اختلاف مقیاس به طور مستقیم بر روی معیارهای خطا تأثیر گذاشته و باعث شده است که مقادیر خطا در مقاله بسیار بزرگ‌تر از مقادیر ما باشند. با در نظر گرفتن این موضوع، نتایج حاصل از مدل‌ها به تفکیک مقایسه می‌شوند.

در ابتدا به معیار MAE یا میانگین خطای مطلق می‌پردازیم. در مقاله، بهترین عملکرد مربوط به مدل LSTM است که MAE برابر با ۶۶۷,۶۳ دارد. مدل GRU نیز با MAE برابر با ۵۳۱,۰۵ عملکرد خوبی نشان داده است، در حالی که Bi-LSTM MAE برابر با ۸۷۸,۴۱ دارد. در مقابل، در پروژه ما، مدل LSTM با MAE برابر با ۱,۵۹، GRU با MAE برابر با ۱,۱۲ و Bi-LSTM با MAE برابر با ۴۸.۱ به نتایجی دست یافته‌اند که اگرچه از نظر عددی کوچک‌تر هستند، اما به دلیل مقیاس متفاوت داده‌ها نمی‌توان مستقیماً با اعداد مقاله مقایسه شوند. با این حال، در مقیاس داده‌های ما، این مقادیر نشان‌دهنده عملکرد مناسبی هستند.

برای معیار RMSE یا ریشه میانگین مربعات خطا، در مقاله مقدار RMSE برای مدل LSTM برابر با ۸۹۶,۷۸ گزارش شده است، در حالی که برای GRU برابر با ۷۱۸,۰۰ و برای Bi-LSTM برابر با ۱،۱۰۸,۹۱ است. در پروژه ما، RMSE برای LSTM برابر با ۲,۲۴، برای GRU برابر با ۱,۶۱ و برای Bi-LSTM برابر با ۲,۰۳ محاسبه شده است. مجدداً، این تفاوت‌ها به مقیاس داده‌ها برمی‌گردد و نمی‌توان اعداد را بدون در نظر گرفتن مقیاس یکسان مقایسه کرد. اما عملکرد مدل‌های ما در مقیاس خودشان بسیار دقیق و مناسب بوده است.

معیار R-Squared یا ضریب تعیین، یکی دیگر از معیارهای مهم است که میزان تطابق پیش‌بینی‌ها با مقادیر واقعی را نشان می‌دهد. در مقاله، مقدار R-Squared برای LSTM برابر با ۰,۹۴۲۵ و برای GRU برابر با ۰,۹۶۳۱ گزارش شده است. این اعداد نشان‌دهنده عملکرد قوی مدل‌ها در مقاله هستند. در پروژه ما، R-Squared برای LSTM برابر با ۰,۹۸۸، برای GRU برابر با ۰,۹۸۹ و برای Bi-LSTM برابر با ۰,۹۸۴ است. این نتایج نشان می‌دهند که مدل‌های ما از نظر تطابق نسبی حتی بهتر از مدل‌های مقاله عمل کرده‌اند. این موضوع احتمالاً به دلیل کیفیت داده‌های مورد استفاده یا پارامترهای بهینه‌سازی‌شده مدل‌ها در پروژه ما بوده است.

در نهایت، به معیار MAPE یا میانگین درصد خطای مطلق می‌پردازیم. در مقاله، MAPE برای LSTM برابر با ۳,۹۴٪، برای GRU برابر با ۳,۰۵٪ و برای Bi-LSTM برابر با ۵,۲۴٪ است. این مقادیر نشان‌دهنده درصد خطای نسبتاً پایینی در پیش‌بینی‌ها هستند. در پروژه ما، MAPE برای LSTM برابر با ۱۶,۱۲٪، برای GRU برابر با ۱۴,۶۸٪ و برای Bi-LSTM برابر با ۱۹,۲۶٪ است. تفاوت در مقادیر MAPE احتمالاً ناشی از تفاوت در ماهیت و نویز داده‌های مورد استفاده در پروژه ما است. این تفاوت می‌تواند به دلیل چالش‌های بیشتری که در داده‌های ما وجود داشته، تأثیرگذار باشد.

پس از دریافت نتایج مکفی از این بخش ها یک بار نیز کل دادگان تست به یکباره توسط مدل SARIMA پس از یافتن پارامترهای بهینه انجام شد که به دلیل آنکه به یکباره کل بازه تست پیش بینی میگردد این مدل دارای دقت کمتری نسبت به مدل های دیگر بود اما صرفا بعنوان یک تست و خارج از دستور کار پروژه پیاده سازی گردید.